



**Trinity College Dublin**

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

SCHOOL OF COMPUTER SCIENCE AND STATISTICS

**A SEMANTICALLY ORIENTED  
FEATURE ENGINEERING APPROACH  
FOR MULTI-CLASS IMBALANCED  
TEXT CLASSIFICATION**

BANDEH ALI TALPUR

MARCH, 2026

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

## **Declaration**

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open-access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: Bandeh Ali Talpur

# Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Declan O’Sullivan, for his invaluable guidance and support throughout my PhD journey. His expertise in the field of computer science and his dedication to advancing my research has been instrumental in shaping this thesis. I am grateful for his insightful feedback, constructive criticism, and patient mentorship, which have challenged me to develop my ideas and improve the quality of my work.

I would also like to thank the members of my thesis committee for their time and valuable feedback. Their expertise and constructive criticism were instrumental in shaping this thesis. Furthermore, I am grateful to the staff and students of the ADAPT Centre for their assistance and support throughout my studies. Their feedback and encouragement were invaluable in shaping my research ideas and motivating me to achieve my goals.

Finally, I would like to express my gratitude to my family and friends for their unwavering support and encouragement throughout my academic journey. Their love and support have been a constant source of inspiration, and I am grateful for the sacrifices they have made to help me achieve my dreams.

Thank you all.

# Abstract

This thesis investigates the problem of multi-class imbalance in text classification, where conventional learning approaches tend to favour majority classes and often fail to represent minority-class patterns effectively. This limitation is especially important in real-world textual applications, where minority classes may contain the most informative, sensitive, or socially significant signals. To address this problem, the thesis proposes STREAM (Semantically Oriented Vector Based Model), a feature-engineering methodology that combines Pointwise Mutual Information and Semantic Orientation to construct class-sensitive document representations. In the cyberbullying setting, the model is further enriched with predicted user-related features to incorporate contextual information that may support more effective discrimination between classes.

The empirical investigation is carried out using a unified, fold-safe evaluation framework designed to avoid data leakage and support fair comparison across competing approaches. The updated methodology uses nested stratified cross-validation for model selection and performance estimation, followed by separate stratified hold-out evaluation to assess generalisation on unseen data. Weighted F1 and Cohen's kappa are used as the principal evaluation measures, supported by additional metrics for complementary interpretation. In settings where the full optimisation procedure was computationally prohibitive across all datasets and classifier combinations, a standardised emergency-mode protocol was applied to reduce optimisation complexity while maintaining methodological consistency, class stratification, and strict separation between training, validation, and testing stages.

The proposed methodology is evaluated on multiple heterogeneous datasets, including cyberbullying severity detection, New York Times emotional tone classification, Yelp review rating prediction, Reuters topic classification, and 20-Newsgroups categorisation. The results show that semantically oriented feature construction provides a meaningful and interpretable contribution to imbalanced text classification, although the magnitude of improvement varies across domains. STREAM performs particularly strongly on the Cyberbullying, Reuters, and 20-Newsgroups datasets, where it achieves the best or among the most stable results under the reported evaluation settings. On the New York Times and Yelp datasets, STREAM remains competitive and consistently improves over Word2Vec-based alternatives, while strong Bag-of-Words baselines, particularly BoW+SVM, remain highly effective and in some cases superior. These findings indicate that semantically informed feature engineering is especially beneficial when class distinctions are closely aligned with underlying semantic structure, while also showing that strong lexical baselines continue to offer robust performance in some domains.

Overall, the thesis demonstrates that semantically oriented feature engineering offers a robust, interpretable, and practically useful approach to multi-class imbalanced text classification. More broadly, it highlights the importance of rigorous evaluation design, leakage-free experimentation, and careful methodological control when assessing classification models on imbalanced textual data.

# Contents

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| <b>Abstract</b>                                                     | <b>iii</b> |
| <b>Table of Contents</b>                                            | <b>v</b>   |
| <b>List of Figures</b>                                              | <b>xii</b> |
| <b>List of Tables</b>                                               | <b>xv</b>  |
| <b>List of Abbreviations</b>                                        | <b>xix</b> |
| <b>1 Introduction</b>                                               | <b>1</b>   |
| 1.1 Research Overview . . . . .                                     | 1          |
| 1.2 Background and Importance of Multi-Class Imbalance . . . . .    | 2          |
| 1.3 Approaches to Class Imbalance and Their Limitations . . . . .   | 3          |
| 1.4 Motivation for a Feature Engineering-Focused Approach . . . . . | 4          |
| 1.5 Research Aims and Objectives . . . . .                          | 5          |
| 1.6 Research Questions . . . . .                                    | 6          |
| 1.7 Contributions and Scope . . . . .                               | 7          |
| 1.7.1 List of Publications from This Research . . . . .             | 7          |
| 1.7.2 Major Contributions . . . . .                                 | 7          |

|          |                                                                                                        |           |
|----------|--------------------------------------------------------------------------------------------------------|-----------|
| 1.7.3    | Minor Contributions . . . . .                                                                          | 8         |
| 1.7.4    | Limitations and Future Directions . . . . .                                                            | 9         |
| 1.8      | Thesis Layout . . . . .                                                                                | 9         |
| <b>2</b> | <b>Literature Review</b>                                                                               | <b>11</b> |
| 2.1      | Background-Class Imbalance . . . . .                                                                   | 12        |
| 2.2      | Theoretical background of classification algorithms and their performance on imbalanced data . . . . . | 13        |
| 2.3      | Methods to address class imbalance in text classification . . . . .                                    | 14        |
| 2.3.1    | Preprocessing . . . . .                                                                                | 14        |
| 2.3.2    | Text Representation . . . . .                                                                          | 16        |
| 2.3.3    | Classification Algorithms . . . . .                                                                    | 21        |
| 2.3.4    | Data and Algorithm Level Approaches . . . . .                                                          | 24        |
| 2.3.5    | Sampling Techniques . . . . .                                                                          | 26        |
| 2.3.6    | Synthetic Minority Over-sampling Technique . . . . .                                                   | 32        |
| 2.3.7    | Feature Selection . . . . .                                                                            | 33        |
| 2.3.8    | Model Performance Evaluation . . . . .                                                                 | 39        |
| 2.4      | STREAM - Related work . . . . .                                                                        | 42        |
| 2.5      | Chapter Summary . . . . .                                                                              | 44        |
| <b>3</b> | <b>STREAM Methodology</b>                                                                              | <b>47</b> |
| 3.1      | Introduction . . . . .                                                                                 | 47        |
| 3.2      | Understanding STREAM . . . . .                                                                         | 48        |
| 3.3      | Methodological Framework . . . . .                                                                     | 50        |

|       |                                                                    |    |
|-------|--------------------------------------------------------------------|----|
| 3.3.1 | Data Acquisition and Preprocessing . . . . .                       | 50 |
| 3.3.2 | Feature Engineering Using PMI and SO . . . . .                     | 51 |
| 3.3.3 | How PMI-SO Features Mitigate Class Imbalance . . . . .             | 62 |
| 3.3.4 | Visualization and Empirical Validation . . . . .                   | 63 |
| 3.3.5 | Focus on Document-Level Analysis . . . . .                         | 66 |
| 3.3.6 | Handling Multi-Class Classification . . . . .                      | 66 |
| 3.3.7 | Classifier Selection . . . . .                                     | 68 |
| 3.3.8 | Feature Normalization and Scalability Considerations . . . . .     | 69 |
| 3.4   | STREAM Pseudocode . . . . .                                        | 70 |
| 3.5   | Technical Discussion on Methodological Choices . . . . .           | 72 |
| 3.5.1 | Justification for Using PMI and SO . . . . .                       | 72 |
| 3.5.2 | Creation of Feature Vectors Using PMI-SO . . . . .                 | 72 |
| 3.5.3 | Emphasis on Feature Engineering Over External Embeddings . . . . . | 73 |
| 3.5.4 | Rationale for Classifier Choices and Parameters . . . . .          | 73 |
| 3.5.5 | Positioning Against State-of-the-Art Approaches . . . . .          | 76 |
| 3.6   | Performance Evaluation . . . . .                                   | 76 |
| 3.6.1 | Experimental Execution Protocol . . . . .                          | 77 |
| 3.6.2 | Cross-Validation and Model Selection Strategy . . . . .            | 78 |
| 3.6.3 | Holdout Evaluation . . . . .                                       | 79 |
| 3.6.4 | Evaluation Metrics . . . . .                                       | 80 |
| 3.6.5 | Aggregation, Variability, and Comparative Analysis . . . . .       | 81 |
| 3.6.6 | Evaluation Integrity and Limitations . . . . .                     | 81 |
| 3.7   | Conclusion . . . . .                                               | 82 |

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>4</b> | <b>Cyberbullying Severity Detection</b>                               | <b>85</b> |
| 4.1      | Data Input . . . . .                                                  | 86        |
| 4.1.1    | Data Accessibility, Collection, and Annotation . . . . .              | 86        |
| 4.1.2    | Manual Data Annotation . . . . .                                      | 87        |
| 4.1.3    | Preprocessing . . . . .                                               | 89        |
| 4.1.4    | Feature Extraction . . . . .                                          | 90        |
| 4.2      | Feature Generation . . . . .                                          | 90        |
| 4.3      | Feature Engineering . . . . .                                         | 91        |
| 4.3.1    | Network-Based Features . . . . .                                      | 91        |
| 4.3.2    | Activity Features . . . . .                                           | 92        |
| 4.3.3    | User Features . . . . .                                               | 92        |
| 4.3.4    | Content-Based Features . . . . .                                      | 94        |
| 4.3.5    | Personality Features . . . . .                                        | 94        |
| 4.4      | Feature Space Construction and Predictive Feature Integration . . . . | 95        |
| 4.5      | Experimental Setup and Emergency-Mode Evaluation . . . . .            | 96        |
| 4.6      | Results and Analysis . . . . .                                        | 97        |
| 4.6.1    | Class distribution and imbalance implications . . . . .               | 97        |
| 4.6.2    | Emergency-mode nested cross-validation results . . . . .              | 97        |
| 4.6.3    | Holdout evaluation and confidence intervals . . . . .                 | 99        |
| 4.6.4    | Ten-fold cross-validation results . . . . .                           | 100       |
| 4.6.5    | Stress tests: keep=0.6 and keep=0.4 . . . . .                         | 100       |
| 4.6.6    | Scenario-level stress summaries . . . . .                             | 102       |
| 4.6.7    | Hypothesis tests . . . . .                                            | 103       |

|          |                                                             |            |
|----------|-------------------------------------------------------------|------------|
| 4.6.8    | Confusion matrices and class-wise error structure . . . . . | 104        |
| 4.6.9    | Critical synthesis . . . . .                                | 104        |
| 4.7      | Domain Analysis - Cyberbullying . . . . .                   | 107        |
| 4.8      | Chapter Summary . . . . .                                   | 111        |
| <b>5</b> | <b>Evaluating STREAM Across Diverse Textual Domains</b>     | <b>113</b> |
| 5.1      | Introduction . . . . .                                      | 113        |
| 5.1.1    | Datasets and Their Characteristics . . . . .                | 114        |
| 5.1.2    | Newspaper (NY Times) Dataset . . . . .                      | 114        |
| 5.1.3    | Yelp Reviews Dataset . . . . .                              | 118        |
| 5.1.4    | Reuters Dataset . . . . .                                   | 119        |
| 5.1.5    | 20 Newsgroups Dataset . . . . .                             | 121        |
| 5.2      | Experimental Setup . . . . .                                | 123        |
| 5.2.1    | Baseline Approaches and STREAM . . . . .                    | 123        |
| 5.2.2    | Parameter Tuning and Protocol . . . . .                     | 123        |
| 5.2.3    | Evaluation Metrics and Reporting . . . . .                  | 127        |
| 5.3      | Results and Analysis . . . . .                              | 127        |
| 5.3.1    | Newspaper (NY Times) Results . . . . .                      | 127        |
| 5.3.2    | Yelp Results and Analysis . . . . .                         | 135        |
| 5.3.3    | Reuters Results and Analysis . . . . .                      | 145        |
| 5.3.4    | 20-Newsgroups Results and Analysis . . . . .                | 153        |
| 5.4      | Summary . . . . .                                           | 159        |

|          |                                                                    |            |
|----------|--------------------------------------------------------------------|------------|
| <b>6</b> | <b>Comparative Analysis Across Modelling Approaches</b>            | <b>161</b> |
| 6.1      | Comparative Evaluation Context . . . . .                           | 161        |
| 6.2      | Why Representation-Level Evidence Is Central to Comparison . . . . | 162        |
| 6.3      | Cluster Analysis as an Interpretive Methodology . . . . .          | 162        |
| 6.3.1    | Sub-Cluster Structure and Minor Disjuncts . . . . .                | 163        |
| 6.4      | t-SNE Projections and Local Separability . . . . .                 | 163        |
| 6.4.1    | Cyberbullying Dataset . . . . .                                    | 164        |
| 6.4.2    | Yelp Dataset . . . . .                                             | 165        |
| 6.4.3    | Newspaper Dataset . . . . .                                        | 166        |
| 6.4.4    | Comparative Interpretation . . . . .                               | 167        |
| 6.5      | Cross-Dataset Comparative Behaviour of STREAM . . . . .            | 168        |
| 6.5.1    | Alignment Between STREAM and Dataset Structure . . . . .           | 168        |
| 6.5.2    | Persistence of Strong Lexical Baselines . . . . .                  | 168        |
| 6.5.3    | Word2Vec as a Comparative Baseline . . . . .                       | 169        |
| 6.6      | Robustness Under Imbalance Stress . . . . .                        | 169        |
| 6.7      | Classifier Interactions and Decision Boundaries . . . . .          | 169        |
| 6.8      | Synthesis and Relation to Thesis Contribution . . . . .            | 170        |
| <b>7</b> | <b>Conclusions and Future Work</b>                                 | <b>171</b> |
| 7.1      | Overview . . . . .                                                 | 171        |
| 7.1.1    | Main Research Question and Sub-Questions . . . . .                 | 171        |
| 7.1.2    | Limitations . . . . .                                              | 173        |
| 7.2      | Future Directions . . . . .                                        | 175        |

|     |                         |     |
|-----|-------------------------|-----|
| 7.3 | Final Remarks . . . . . | 176 |
|-----|-------------------------|-----|



# List of Figures

|     |                                                                                                                         |     |
|-----|-------------------------------------------------------------------------------------------------------------------------|-----|
| 3.1 | An overview of the STREAM methodology highlighting core and optional steps. . . . .                                     | 48  |
| 3.2 | Visualization of Documents Using PMI-SO Features and PCA . . . .                                                        | 65  |
| 4.1 | Distribution of Tweets by Cyberbullying Severity . . . . .                                                              | 89  |
| 4.2 | Cyberbullying Severity: Holdout confusion matrices for leading BoW-based models. . . . .                                | 105 |
| 4.3 | Cyberbullying Severity: Confusion matrices under keep=0.6 stress testing across all approaches and classifiers. . . . . | 106 |
| 4.4 | A treemap with rectangle sizes and colors determined by each feature's odds ratio . . . . .                             | 109 |
| 5.1 | Class Distribution in NY Times Dataset . . . . .                                                                        | 116 |
| 5.2 | Class Distribution in Yelp Dataset . . . . .                                                                            | 119 |
| 5.3 | Top 10 Categories in Reuters Dataset . . . . .                                                                          | 120 |
| 5.4 | KNN Kernel Tunning . . . . .                                                                                            | 124 |
| 5.5 | SVM Kernel Tunning . . . . .                                                                                            | 125 |
| 5.6 | NY Times: Holdout confusion matrices for leading models . . . . .                                                       | 131 |
| 5.7 | Yelp: Confusion matrices for selected holdout models . . . . .                                                          | 139 |

|     |                                                                                                        |     |
|-----|--------------------------------------------------------------------------------------------------------|-----|
| 5.8 | Yelp: Confusion matrices under keep=0.6 stress testing across all approaches and classifiers . . . . . | 142 |
| 5.9 | Yelp: Confusion matrices under keep=0.4 stress testing across all approaches and classifiers. . . . .  | 143 |
| 6.1 | t-SNE plots for three feature representations on the Cyberbullying dataset . . . . .                   | 164 |
| 6.2 | t-SNE plots for three feature representations on the Yelp dataset . . .                                | 167 |
| 6.3 | t-SNE plots for three feature representations on the Newspaper dataset                                 | 167 |

# List of Tables

|     |                                                                                                                                  |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------|-----|
| 2.1 | Studies that have dealt with text classification preprocessing . . . . .                                                         | 25  |
| 2.2 | Confusion Matrix . . . . .                                                                                                       | 39  |
| 2.3 | References to the dataset's literature, as well as examples of text representation methods, classifiers, and imbalance . . . . . | 45  |
| 3.1 | Word Occurrences in Classes . . . . .                                                                                            | 55  |
| 3.2 | Summary of Documents for Visualization . . . . .                                                                                 | 63  |
| 4.1 | Class distribution and imbalance ratios . . . . .                                                                                | 97  |
| 4.2 | Comparative performance summary across approaches and classifiers (emergency-mode nested CV). . . . .                            | 98  |
| 4.3 | Holdout test set performance . . . . .                                                                                           | 99  |
| 4.4 | Comprehensive 10-fold CV metrics . . . . .                                                                                       | 100 |
| 4.5 | Comprehensive metrics under imbalance stress (keep=0.6) . . . . .                                                                | 101 |
| 4.6 | Comprehensive metrics under imbalance stress (keep=0.4) . . . . .                                                                | 102 |
| 4.7 | stress60: Performance metrics across different evaluation scenarios .                                                            | 102 |
| 4.8 | stress40: Performance metrics across different evaluation scenarios .                                                            | 103 |
| 4.9 | Summary of hypothesis tests across evaluation scenarios . . . . .                                                                | 103 |

|      |                                                                                                                           |     |
|------|---------------------------------------------------------------------------------------------------------------------------|-----|
| 5.1  | Best KNN Configurations Across Datasets . . . . .                                                                         | 125 |
| 5.2  | Best SVM Kernel Configurations Across Datasets . . . . .                                                                  | 126 |
| 5.3  | Class distribution for the NY Times dataset. . . . .                                                                      | 128 |
| 5.4  | Comparative performance summary (nested cross-validation, weighted metrics). . . . .                                      | 129 |
| 5.5  | 10-fold cross-validation results (weighted metrics). . . . .                                                              | 129 |
| 5.6  | Dominant confusion patterns for best-performing models on the NY Times dataset. . . . .                                   | 132 |
| 5.7  | Holdout evaluation results with 95% confidence intervals . . . . .                                                        | 133 |
| 5.8  | Hypothesis testing summary for NY Times dataset . . . . .                                                                 | 134 |
| 5.9  | Performance under imbalance stress (keep = 0.6) . . . . .                                                                 | 134 |
| 5.10 | Performance under imbalance stress (keep = 0.4) . . . . .                                                                 | 134 |
| 5.11 | Yelp: Class distribution and imbalance ratios . . . . .                                                                   | 136 |
| 5.12 | Yelp: Comparative nested cross-validation performance summary across approaches and classifiers (3 outer folds) . . . . . | 137 |
| 5.13 | Yelp: Holdout evaluation results with 95% confidence intervals for weighted $F_1$ . . . . .                               | 138 |
| 5.14 | Yelp: Dominant confusion patterns for selected holdout models. . . .                                                      | 140 |
| 5.15 | Yelp: Stress test performance under imbalance perturbation (keep=0.6)                                                     | 144 |
| 5.16 | Yelp: Stress test performance under imbalance perturbation (keep=0.4).                                                    | 144 |
| 5.17 | Yelp: Summary of hypothesis tests across evaluation scenarios. . . .                                                      | 145 |
| 5.18 | Reuters: Class distribution and imbalance ratios. . . . .                                                                 | 146 |
| 5.19 | Reuters: Comparative nested cross-validation performance summary across approaches and classifiers . . . . .              | 147 |

|                                                                                                                                      |     |
|--------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.20 Reuters: Holdout evaluation results with 95% confidence intervals for weighted $F_1$ . . . . .                                  | 148 |
| 5.21 Reuters: Stress test performance under per-class retention reduction (keep=0.6) . . . . .                                       | 150 |
| 5.22 Reuters: Stress test performance under per-class retention reduction (keep=0.4) . . . . .                                       | 150 |
| 5.23 Reuters: Summary of hypothesis tests across evaluation scenarios. . .                                                           | 151 |
| 5.24 Previously Reported Results on the Reuters Dataset (Selected) . . . .                                                           | 152 |
| 5.25 Class distribution and imbalance ratios for the stratified 20-Newsgroups subset and holdout split. . . . .                      | 154 |
| 5.26 Nested cross-validation performance across approaches and classifiers (mean $\pm$ standard deviation over outer folds). . . . . | 155 |
| 5.27 20-Newsgroups: Holdout evaluation results with 95% confidence intervals for weighted $F_1$ . . . . .                            | 155 |
| 6.1 Cyberbullying Sub-Cluster Distribution (K=3). Values are #items (% of that class). . . . .                                       | 164 |
| 6.2 Yelp Sub-Cluster Distribution (K=3). Values are #items (% of that class).165                                                     |     |
| 6.3 Newspaper Sub-Cluster Distribution (K=3). Values are #items (% of that class). . . . .                                           | 166 |



# List of Abbreviations

|      |                              |
|------|------------------------------|
| AUC  | Area Under the Curve         |
| BoW  | Bag-of-Words                 |
| cBoW | continuous Bag-of-Words      |
| DT   | Decision Trees               |
| FN   | False negative               |
| FP   | False Positive               |
| KNN  | K-Nearest Neighbor           |
| NB   | Naïve Bayes                  |
| NLP  | Natural Language Processing  |
| PCA  | Principal Component Analysis |
| PMI  | Pointwise Mutual Information |
| POS  | Part-Of-Speech               |
| RF   | Random Forest                |

|        |                                           |
|--------|-------------------------------------------|
| ROC    | Receiver Operation Characteristic         |
| SMOTE  | Synthetic Minority Oversampling Technique |
| SO     | Semantic Orientation                      |
| STREAM | Semantically Oriented Vector Based Model  |
| SVM    | Support Vector Machine                    |
| TF-IDF | Term Frequency Inverse Document Frequency |
| TN     | True Negative                             |
| TP     | True Positive                             |

# 1 | Introduction

## 1.1 Research Overview

The exponential growth of textual data in digital platforms has intensified the challenges associated with automatic text classification. While traditional machine learning methods have achieved remarkable successes in various domains—from spam detection to sentiment analysis—these approaches often assume balanced class distributions. In reality, data rarely conform to such assumptions. Classes may exhibit pronounced imbalances, with minority classes severely underrepresented. This discrepancy leads to biased model predictions favoring the majority classes and neglecting the minority classes (Santoso et al. 2017; W. Chen et al. 2024). The consequences are particularly problematic in domains where correctly identifying the minority classes is critical, such as medical diagnoses (rare diseases), fraud detection (unusual but crucial events), and content moderation tasks like cyberbullying detection (identifying the most harmful or severe incidents) (Haixiang et al. 2017; Talpur et al. 2020; Z. Yang et al. 2022).

Though class imbalance has been explored extensively in binary classification problems, multi-class imbalance introduces additional layers of complexity. Instead of merely distinguishing between one majority and one minority class, multi-class scenarios may have multiple minority classes, multiple majority classes, or intricate distributions where one or more classes are severely skewed (Lango 2019). Such complexity renders straightforward adaptations of binary-focused imbalance tech-

niques less effective. Methods that excel in binary contexts may fail to scale or adapt well to the multi-class domain (Phoungphol 2013), leading to suboptimal performance and missed opportunities to glean value from the minority classes—often the ones carrying the most critical signals (Leevy et al. 2018).

This thesis directly confronts the issue of multi-class imbalance in text classification. The primary contribution is the creation and assessment of **STREAM (Semantically Orientated Vector Based Model)**, an innovative methodology aimed at reducing high dimensionality, minimising noise, and utilising semantic linkages to improve minority class representation. In contrast to traditional methods that depend exclusively on data-level resampling or algorithmic cost modification, STREAM employs a feature engineering-focused approach, incorporating semantic orientation (derived from Pointwise Mutual Information, PMI) and user-related predicted features to generate enhanced, contextually significant embeddings. This method seeks to enhance performance in both multi-class situations, involving several minority and majority classes, and in binary contexts, hence ensuring wide applicability across diverse data distributions and complexities of problems.

## 1.2 Background and Importance of Multi-Class Imbalance

The recent emergence of new technologies, such as the internet of things and online social networks, has facilitated the generation of vast amounts of data known as Big Data. This influx presents challenges for knowledge extraction through learning algorithms, primarily due to its distinct characteristics: variety, volume and, velocity. Furthermore, the issue of imbalanced data is commonly encountered in these types of datasets. When taking into account the multi-class nature, the challenge of learning from large, multi-class imbalanced datasets becomes significant and remains underexplored in existing studies, despite its frequent occurrence in real-world classification scenarios (Tanha et al., 2020).

Multi-class imbalance is not merely a theoretical inconvenience; it frequently occurs in real-world data. In the identification of cyberbullying intensity, the majority of posts may be innocuous or merely somewhat hostile, whereas genuinely serious harassment represents a negligible portion of the dataset. Likewise, in news categorization or company rating prediction, certain categories or ratings may be relatively infrequent, complicating the model's ability to learn robust patterns for these underrepresented classes. Addressing multi-class imbalance is essential for practical application; models must generalize effectively across all classes, not just the predominant ones.

### **1.3 Approaches to Class Imbalance and Their Limitations**

A range of strategies have been proposed to counter class imbalance, generally falling into two broad categories (A. Ali et al. 2013):

Data-level methods include data sampling (under-sampling and over-sampling) and feature selection approaches, while algorithm-level methods include cost-sensitive and hybrid/ensemble approaches (Leevy et al. 2018). Data-level methods include various resampling techniques such as synthetic minority over-sampling technique (SMOTE) (Bach et al. 2019), which manipulate training data directly and rebalance skewed class distributions to decrease the level of imbalance or noise reduction, like random under-sampling (remove random samples from the majority class), random oversampling (doubles random samples from the minority class), and combinations of these. Although these methods are simple to apply, they may have overfitting and over-generalization issues. By selectively eliminating data, under-sampling lowers the overall quantity of information for the model to learn. Oversampling frequently results in longer training times due to larger training data sets and overfitting. A model that closely matches the training data and is unable to generalise to the new data is said to be overfitting (high variation in data) (Johnson et al. 2019).

The issue of class imbalance is critically important since it occurs in significant domains and, in certain situations, significantly hinders the effectiveness of traditional learning methods that assume a balanced class distribution (Japkowicz 2000). The class imbalance issue in machine learning on various data types has so far been addressed in several ways (Johnson et al. 2019). However, a significant portion of earlier text classification research has concentrated in particular on class imbalance in binary instances (Ah-Pine et al. 2016; Akkaradamrongrat et al. 2019; Ghosh et al. 2019; Y. Li et al. 2010; Madabushi et al. 2020; Mladenic et al. 1999; J. Yang et al. 2014; Y. Zhao et al. 2018). In contrast, very few research have looked at the issue of class imbalance in situations involving multi-class scenarios (Damaschk et al. 2019; Lango 2019; Zheng et al. 2004). However, several research have offered solutions for instances including both binary and multi-class scenarios (E. Chen et al. 2011; Padurariu et al. 2019; Yanmin Sun et al. 2009).

## **1.4 Motivation for a Feature Engineering-Focused Approach**

Feature engineering provides a unique angle on the imbalance problem. Instead of merely resampling or adjusting costs, engineering features that highlight the minority classes’ semantic and contextual cues can drastically improve minority class detection. This idea underlies the STREAM methodology introduced in this thesis.

STREAM not only tackles class imbalance but also incorporates semantic orientation (SO) generated from PMI, effectively capturing the intricate semantic structure of text. STREAM enhances models’ ability to distinguish between classes by converting raw textual input into semantically rich embeddings that highlight aspects pertinent to minority classes, even in the presence of significant imbalance. Furthermore, by integrating anticipated user-related characteristics (e.g., Gender, Age, Personality in the context of social media data), STREAM utilizes domain context and user behavior patterns that may be associated with minority-class cases.

## 1.5 Research Aims and Objectives

The primary goal of this research is to design, implement, and evaluate the STREAM methodology to address multi-class imbalance in text classification. The key objectives include:

- **Identify and analyze multi-class imbalance issues** in diverse real-world text datasets, understanding their impact on classification performance and minority class detection.
- **Develop the STREAM methodology**, focusing on feature engineering through PMI-based semantic orientation, predicted user attributes, and potentially class imbalance handling techniques like SMOTE, to create vector-like embeddings that improve the representation of minority classes.
- **Thoroughly evaluate STREAM** on multiple datasets:
  1. **Cyberbullying dataset:** As a complex multi-class scenario with severity-level-based classes and severe imbalance issues, this dataset tests STREAM's ability to capture rare but critical harassment patterns.
  2. **Newspaper tweets (NY Times):** Containing emotional tone categories, this dataset checks STREAM's adaptability to a different textual domain and a different form of multi-class imbalance, validating that STREAM is not confined to a single data type or context.
  3. **Yelp reviews (rating prediction), 20-Newsgroups, and Reuters datasets:** By extending evaluation to widely used benchmark corpora and different classification tasks (such as sentiment-based rating prediction or topical categorization), STREAM's robustness, generality, and scalability can be demonstrated.
- **Compare STREAM against baseline approaches** (Bag-of-Words, Word2Vec) and existing class imbalance solutions, using multiple metrics (Kappa, F-Measure,

ROC for binary tasks) and validation strategies (including 10-fold cross validation on the 80% training portion and final evaluation on a 20% test set) to ensure reliable and generalizable performance.

- **Examine binary and multi-class scenarios:** Although the primary emphasis lies in multi-class imbalance, demonstrating STREAM’s effectiveness in binary contexts further underscores its practical utility and broad applicability.

## 1.6 Research Questions

The central research question guiding this thesis is:

*To what extent can the STREAM methodology improve machine learning classifier performance when addressing multi-class imbalances in text classification?*

This question can be further broken down into sub-questions:

1. How do multiple minority or majority classes affect classifier training? Which scenario proves more challenging?
2. Which combination of data-level and algorithm-level approaches, integrated via feature engineering, best mitigates multi-class imbalances in text classification?
3. How does STREAM’s semantic and user-based feature integration impact performance in both multi-class and binary imbalance scenarios?
4. Can STREAM maintain its advantages when the number of available features is limited, reflecting more constrained real-world conditions?

## 1.7 Contributions and Scope

### 1.7.1 List of Publications from This Research

#### 1. Cyberbullying severity detection: A machine learning approach

*B. A. Talpur, D. O'Sullivan*

*PloS One 15 (10), e0240924*

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0240924>

**Citations:** 139 (as recorded February 2026)

*Summary:* This publication showcases the proposed methodology (STREAM) applied to an open-source Twitter dataset in the domain of harassment. Although the dataset had some limitations regarding the provided features, the paper reports two significant contributions: (1) it demonstrates the better performance achieved by the STREAM approach, and (2) it introduces the concept of detecting severity in the cyberbullying dataset.

#### 2. Multi-class imbalance in text classification: A feature engineering approach to detect cyberbullying in Twitter

*B. A. Talpur, D. O'Sullivan*

*Informatics 7 (4), 52*

<https://www.mdpi.com/2227-9709/7/4/52>

**Citations:** 44 (as recorded February 2026)

*Summary:* In this publication, the authors delve into a cyberbullying dataset collected from Twitter and manually labeled. This paper offers an in-depth analysis of the severity of cyberbullying through the STREAM methodology.

### 1.7.2 Major Contributions

- **The STREAM Methodology:** A novel feature-engineering approach for addressing multi-class imbalance in text classification by incorporating semantic

orientation and predicted features. STREAM enhances minority class visibility without excessive reliance on traditional oversampling or cost-sensitive methods.

- **Broad Evaluation on Multiple Datasets:** Through experiments on the cyberbullying dataset (multi-class severity), newspaper tweets (emotional tone classification), Yelp reviews (rating prediction), and benchmark corpora (20-Newsgroups, Reuters), STREAM’s versatility and robustness are demonstrated.
- **Integration of Predicted Features:** By incorporating predicted user-related attributes in scenarios like cyberbullying severity detection, STREAM highlights how domain knowledge and contextual signals can further mitigate class imbalance.
- **Improved Metric Use and Validation Protocols:** Using Kappa, F-Measure, and ROC metrics and a stringent evaluation strategy (including cross-validation and final test splits) ensures that reported improvements are robust, not artifacts of overfitting or chance.

### 1.7.3 Minor Contributions

- Demonstration that PMI-driven semantic features can outperform standard text representations (e.g., Bag-of-Words, Word2Vec) in imbalanced settings.
- Introduction of multi-class severity detection in cyberbullying—a novel angle that reveals the complexity of minority class identification in sensitive domains.
- Exploration of emotion-laden lexical resources (e.g., WDAL in newspaper datasets) within STREAM, showing how domain-specific lexical tools can integrate into the pipeline.

### 1.7.4 Limitations and Future Directions

While STREAM shows promise across diverse datasets, it may require substantial computational resources and might not always excel in domains with sparse semantic cues. Additionally, expanding beyond the examined datasets and incorporating more sophisticated or hybrid imbalance methods could further validate STREAM’s universal applicability. Future work may involve extending STREAM to non-textual modalities, integrating more domain-specific knowledge, or refining the feature selection processes with advanced neural embeddings and transformer-based architectures.

## 1.8 Thesis Layout

**Chapter 1** (This Chapter): Introduces the problem of multi-class imbalance in text classification, motivates the need for novel solutions, outlines the research questions and objectives, and presents the STREAM methodology’s conceptual foundation.

**Chapter 2** Reviews the literature on class imbalance approaches, text classification methods, and feature selection strategies. It situates the STREAM methodology within the broader research landscape.

**Chapter 3** Details the STREAM methodology: its theoretical underpinnings, the steps for PMI-based semantic orientation, predicted feature integration, and how it interfaces with various classifiers.

**Chapter 4** Applies STREAM to the cyberbullying dataset, addressing a real-world, highly imbalanced multi-class scenario, and compares results to baseline methods.

**Chapter 5** Extends evaluation to newspaper (emotional tone), Yelp, 20-Newsgroups, and Reuters datasets, confirming STREAM’s adaptability and generalization capabilities.

**Chapter 6** Provides an in-depth, data-level analysis of how Bag-of-Words, Word2Vec, and STREAM handle nuances like sub-clusters, ambiguous samples, and high-confidence errors in three representative datasets. This comparative exploration offers deeper insights into *why* STREAM achieves its observed performance gains and how each method copes with imbalanced data complexities.

**Chapter 7** Concludes the thesis by summarizing the key contributions, limitations, and potential avenues for enhancing and applying STREAM in broader contexts.

## 2 | Literature Review

Text classification is a crucial activity in natural language processing (NLP) that includes categorising text material into predefined categories. With an emphasis on resolving class imbalance, this part of the literature review offers an overview of the various methods and approaches used in text categorization. The research gap is identified by understanding state-of-the-art approaches used to address class imbalance in text classification. The chapter highlights the steps involved in the text classification process, such as sorting text, sentiment analysis, topic analysis, question answering, natural language inference, text representation, and classification algorithms. Additionally, it goes through a number of methods and approaches to deal with the major difficulties in text categorization, such data efficiency and class imbalance. The strategies discussed include sampling techniques such as over-sampling, under-sampling, and blended approaches, SMOTE, feature selection, and cost-sensitive approach. Additionally, the chapter discusses text representation techniques such as Bag of Words (BoW) and Word2Vec, which play a crucial role in text classification. Furthermore, the chapter highlights model performance evaluation techniques, including single-class performance measures, overall performance measures, and Kappa. Understanding these techniques and strategies can help in the development of efficient and accurate text classification models, especially in addressing class imbalance. Overall, this literature review chapter provides valuable insights into the various techniques and strategies used in text classification, specifically in addressing class imbalance. By identifying the research gap, this review provides a foundation for future research in this area.

## 2.1 Background–Class Imbalance

The term "class imbalance" refers to a dataset with a majority and minority class. Class imbalances can vary from "slightly imbalanced" to "rarity," in accordance to the spectrum. Where Small numbers of positive examples are related to dataset rarity (Bauder et al. 2018). Data scientists often use the binary technique for analytics since approach may simplify many multi-class situations. In real-world situations, the minority (positive) class, which makes up a smaller proportion of the dataset, is frequently the class of interest (Leevy et al. 2018), whereas the greater percentage is made up of the majority (negative) class (Hasanin et al. 2019).

The class imbalance issue has received a lot of attention from academics in the most recent ten years since it poses a significant obstacle to the disciplines of machine learning and data mining. A classifier's overall accuracy would be excellent for a given dataset with an issue of class imbalance, but its performance on the minority class would be poor. There are two categories of data where this issue may arise: (a) binary problems in which one of two classes has a disproportionately large number of samples, and (b) multi-class problems in which no single class contains more than a negligible number of examples (Wasikowski et al. 2010).

Researchers in the fields of machine learning and data mining are drawn to the class imbalance problem because it presents a challenge in both theoretical and applied settings. The studies included below concentrate on three aspects of the class gap. First, given the problem of class imbalance, how should classification performance be assessed? Second, what factors contribute to class imbalances, and under what conditions do they have the most effects on a regular classifier's effectiveness? (Japkowicz and Stephen 2002). What can be done, thirdly, to solve the social stratification issue? It is mentioned in the opening section that the most popular statistic for assessing prediction models and direct search methods is precision. When evaluating a classification model produced from a set of data with an uneven class distribution, accuracy is no longer a valid criteria since unusual classes have far less ef-

fect on precision than regular classes (Joshi et al. 2001). Several alternative evaluation measures, such as precision, recall, G-mean, F-measure, and a Receiver Operation Characteristic (ROC) Curve Analysis, are then explored and recommended. Most of the offered strategies are limited to a binary classification setting (Y. Sun et al. 2006).

The imbalanced problem occurs in bi-class applications when there are disproportionately many samples from one class than the other (Wasikowski et al. 2010). Although the class imbalance issue is most pronounced in a bi-class society, it is not unique to that setting. Multi-class instances, which account for the vast majority of real-world applications, suffer from subpar classification results due to uneven class distributions (Y. Sun et al. 2006). As data imbalance happens naturally in data, it is a problem for many real-world machine learning systems. Sentiment and emotion analysis are two fields where the data imbalance problem is especially acute. The performance of sentiment and emotion classification faces significant challenges due to imbalanced training data (Suhaeni et al. 2023).

## **2.2 Theoretical background of classification algorithms and their performance on imbalanced data**

Theoretical foundations of classification algorithms and how they operate on imbalanced data play a vital role in addressing these challenges.

One of the key factors that affect classification performance on imbalanced data is the algorithm's bias-variance trade-off. A key idea in supervised machine learning studies is the bias-variance trade-off. It claims that the trade-off between an algorithm's complexity (variance) and bias determines how well it generalises the new data. Contrary to variance, which describes the error brought on by overly complex models, bias describes the inaccuracy brought on by oversimplifying the model. Low bias algorithms are more likely to overfit the training set since they are often more complex (i.e., have larger variance). On the other hand, a high bias or

low variance algorithm could be too simple and more likely to underfit the training set of data (Rashidi et al. 2023).

To address these challenges, various theoretical frameworks and modifications to classification algorithms have been proposed. For instance, some researchers have proposed using modified cost functions that penalize misclassification of the minority class more heavily than the majority class (Charles X. Ling and C. Li 1998). Other scholars have suggested adjusting the algorithm's determination threshold to strike a balance between the value of recall and accuracy (Batista et al. 2004).

In general, establishing practical solutions to class imbalance in text classification requires a grasp of the theoretical foundations of classification algorithms and their performance on data that is imbalanced. Researchers can create novel methods and adjustments that enhance classification performance on imbalanced datasets by understanding the fundamental concepts of these algorithms.

## **2.3 Methods to address class imbalance in text classification**

### **2.3.1 Preprocessing**

Text preprocessing, text representation or transformation, dimensionality reduction, selection and application of classification techniques, classifier evaluation, and classifier validation are the six interconnected steps that make up the text classification process (Kowsari et al., 2019). Preprocessing may be used to improve a dataset's quality in general and for text categorization in particular. Within this particular context, the term "preprocessing" pertains to the systematic procedure of purifying and organising textual data in order to facilitate its subsequent classification (Avasthi et al. 2022). Hence, preprocessing plays a vital role both before and during the feature selection process, and its significance cannot be overstated. It not only eliminates noise and emphasizes key features but also significantly reduces the

time needed for training and testing a model (Siino et al. 2024) Converting uppercase letters to lowercase letters, correcting common misspellings, removing HTML tags and punctuation marks, reducing the number of duplicate characters, reducing the number of emoticon labels to a reduced set of wildcard characters, reducing the number of stop words, replacing HTTP links with wildcard characters, removing stop words, and word stemming are all examples of types that fall into the most basic category of preprocessing (Avasthi et al. 2022; HaCohen-Kerner, Miller, et al. 2020).

Text preprocessing is a widely studied area in NLP machine learning. Many recent studies have shown the importance of text preprocessing for classification tasks and the impact it can have on classification performance. For example, Uysal et al. (2014), found that text preprocessing techniques such as stemming, stop word removal, and feature selection can significantly improve the accuracy of text classification models. Similarly, in a study by (HaCohen-Kerner, Miller, et al. 2020), the authors found that the use of spell checking and conversion into lowercase letters can improve the accuracy of text classification. Other recent research has concentrated on creating new text preprocessing methods for certain text categorization problems. For example, in a study by Zhou et al. (2020), the authors suggested an innovative technique for text preprocessing in hate speech identification that entailed locating and eliminating hazardous terms from text data. These studies, taken together, emphasise the significance of text preprocessing for classification problems and show the potential advantages of adopting sophisticated text preprocessing methods to raise classification accuracy.

Many text categorization systems just perform a handful of elementary preprocessing operations, such as converting all capital letters to lowercase and eliminating stop words. However, the majority of text categorization algorithms don't perform any preprocessing. Furthermore, text categorization researchers have demonstrated that not all preprocessing methods are effective. Some of them could even result in a reduction in ultimate classification accuracy (HaCohen-Kerner, Miller, et al. 2020).

There aren't many articles in the field of text categorization research that attempt to solve the class imbalance. Class imbalance in text-based analysis is a problem that is frequently ignored, and only a few academics have even made an effort to address it (Qiu et al. 2018).

### **2.3.2 Text Representation**

Text classification is the process of assigning texts in the documents to some predefined categories based on their content (Avasthi et al. 2022). It involves identifying the main ideas in sentences and evaluating the viewpoints, objectives, and feelings of the author. Text classification technology can be considered as the foundation of NLP and social network analysis, and it helps with online information finding and public opinion research (R. Li et al. 2022). It is desirable to automate the text categorization process using machine learning techniques to get more accurate and objective findings. By identifying the necessary information, this can also improve the effectiveness of information retrieval and reduce the issue of information overload (Q. Li et al. 2021; Palanivinayagam et al. 2023).

From the 1960s until the 2010s, traditional text classification models remained dominated. Traditional methods mean statistics-based models, such as NB (Maron 1961), KNN (Cover et al. 1967), and SVM (Q. Li et al. 2021). Numerous models have been proposed in the past few decades for text classification. For traditional models, NB (Maron 1961) is the first model used for the text classification task. Whereafter, generic classification models are proposed, where KNN (Cover et al. 1967), SVM (Joachims 1998), and Random Forest (RF) (Breiman 2001) classifiers are widely used for text classification (Q. Li et al. 2021).

The well-known BoW technique to text classification simplifies segmentation by treating each word as a single feature. Although BoW is still helpful, the enormous feature size requires a significant investment in computation nowadays, making feature reduction a crucial preprocessing step. Feature reduction in text classification

often means eliminating characteristics that statistically contribute little to the category. It is possible that feature reduction, in terms of the feature's meaning, could result in reduced classification accuracy. Semantically related words are often treated as distinct features at the current stage of feature reduction, which increases computing complexity and lowers classification accuracy. Several modern technologies may be used to compare the semantic similarity of words. The Word2Vec word embedding technique is gaining popularity because of its low processing expense and excellent accuracy in detecting the semantic similarity of two words. The researchers suggested a technique for lowering the amount of features in a dataset by finding semantically comparable features using a Word2Vec package, then loosely grouping those features using graph search (Ge et al. 2017). Word2Vec employs many models, such as skip-gram and continuous Bag-of-Words (cBoW). The two models have a foundation in the statistical likelihood of adjacent words. By using skip-gram, we can predict which words are most likely to occur after a specific word. cBoW, on the other hand, flips this by predicting which phrases are most likely to arise based on a limited number of context words (Dang et al. 2020).

The statistical measure of how important a word is to a text in a corpus is the term frequency inverse document frequency (TF-IDF). This measure accounts for the word's frequency in the target document and the rest of the corpus. If a term occurs more often in the target document and less often elsewhere, its relevance rises. Typically, TF-IDF is calculated using the vectorizer class in the scikit-learn toolkit. In language modelling and feature learning, the process known as "word embedding" entails giving each word a numerical value that ensures the words with similar meanings are represented in a comparable way, while the process known as "value learning" can be carried out using neural networks (Dang et al. 2020).

### **Bag-of-Words**

BoW works by essentially making a list of all the words that can be found in a given document (Caropreso et al. 2006). The standard topic-based classification model,

which is built on the BoW representation, functions by tying a text to a vector that shows how frequently each chosen word appears in the training corpus. When using a BoW representation, it is generally recommended to use at least one combination of straightforward preprocessing procedures in order to get superior accuracy results (HaCohen-Kerner, Miller, et al. 2020). Characteristics like BoW or bag of  $n$ -grams are typically extracted when using machine learning for NLP (Shao et al. 2018).

The BoW model cannot explain the semantics of a text or the relationships between its words since it believes that words exist in isolation. It is a simple document representation paradigm that has a wide range of applications where its better performance has been demonstrated (Sinoara et al. 2019). In text mining applications, the BoW format is the most straightforward and popular vector text representation (Kucharczyk et al. 2014).

BoW procedures are widely used because they are quick and effective, and they can also achieve extremely high precision. For classifying documents, BoW characteristics can be produced by considering the document as a random collection of words. Despite the effectiveness of BoW and  $n$ -gram approaches, these methods are not without flaws. These methods have a basic flaw in that they fail to account for long-range word ordering, which can result in modifiers and their objects being separated by a huge number of unrelated words (Dai et al. 2015). Since the original wording is no longer there, comparable representations of sentences with the same words but different meanings can be found. Furthermore, as seen by the gaps between the words in an embedding space, BoW characteristics do not fully understand the meaning of the words. Because there is no sense of similarity between words, they are treated as atomic units, resulting in sparse "one-hot" vectors (Barry 2017).

In the study conducted by (HaCohen-Kerner, Miller, et al. 2020), three different supervised machine learning techniques (Bayesian Networks (BN), SVM, and RF),

were employed in the investigation to analyze topic-based text classification. The authors (HaCohen-Kerner, Miller, et al. 2020) used a standard model for text classification, which is the sack of word unigrams. For a variety of text categorization tasks, several earlier research utilising machine learning techniques based on the BoW model that contains hundreds of thousands of unigram words have claimed accuracy rates of 90% or higher. To test the hypothesis that using a wide range of preprocessing strategies could improve text classification outcomes, the authors conducted a series of tests on the time-honored BoW model of 1,000-5,000 unigram words. They also contrasted the effectiveness of the Word2Vec and doc2vec features for text classification with that of the more traditional BoW features. The research produced a range of text categorization findings, and the potential reasons for the variations were carefully examined.

### **Word2Vec**

The BoW representation's limitations can be overcome with the use of more subtle context-sensitive approaches. One example is Word2Vec, a type of word embedding representation created by training a two-layer neural network to associate words with their contexts (Cichosz 2018). In other words, Word2vec is a widely used method for text representation that estimates the probability of a word's occurrence based on the context provided by its neighboring words (Palanivinayagam et al. 2023).

Word embedding techniques have been useful in a number of contexts and have been deemed to be quite beneficial. Word embedding takes the semantics of words into account, allowing different words with similar meanings to be represented by the same or closely related vectors (Palanivinayagam et al. 2023). In particular, the Skip-gram model, developed by (Mikolov, K. Chen, et al. 2013), has helped to popularize these techniques by providing a straightforward, streamlined Neural Network-based approach to generating high-quality vector representations of words from massive amounts of text data. The widely used GloVe technique uses

a global log bilinear regression model to train word embeddings (Pennington et al. 2014).

The application of neural network approaches that directly extract text corpora (i.e., word embeddings) as vectors and then utilise those vectors to represent the words in distributional semantics has garnered interest in this area (Pennington et al. 2014). In the realm of word embeddings research, Word2Vec's notion of cBoW and Skip-gram vector learning models stands out (Mikolov, Sutskever, et al. 2013). By contrasting word embeddings with traditional methods that use co-occurrence counts in a range of lexical semantics tasks, (Baroni et al. 2014) confirmed the potential of word embeddings and concluded that word embeddings are preferred distributional semantic models.

There are several ways to create usable vectors from text owing to the introduction of Word2Vec. Architectures include the cBoW method and the Skip-gram technique. The Skip-gram predicts a word based on its neighbours, whereas the cBoW does the opposite. Studies have shown that the Skip-gram consistently performs better than the cBoW design (Mikolov, Sutskever, et al. 2013). GloVe is one of several Word2Vec variations that have been created, all of which fundamentally accomplish the same. The count-based technique GloVe uses matrix factorization to decrease the problem's dimensionality rather than using a neural network, like Word2Vec does, to learn its word vectors. To replace Word2Vec in the categorization of documents, the doc2vec technique was created (Suqi Jiang et al. 2016). Word2Vec is one of the most useful techniques in neural probabilistic language models. A large number of NLP projects that generate vectors of a defined length to represent words have been completed with state-of-the-art results (Mikolov, Sutskever, et al. 2013). The fundamental premise of Word2Vec is that, given an appropriate training corpus, it should be able to produce a vector space. Every distinct word is then given a vector representation and mapped into the vector space.

In a new take on document representation, Google introduced Word2Vec (Lille-

berg et al. 2015). Word vector representations may be calculated efficiently in a high-dimensional space. Words that share contexts and have comparable interpretations are mapped next to one another in the vector space. In order to retain semantic links in vector space, similarity of word representations get semantic characteristics in addition to syntactic information (Al-Azani et al. 2017). Even though word embeddings have been used in the NLP field for some time, the Word2Vec framework's capacity to swiftly and easily produce vector representations of words has sparked renewed interest in this method. In a low-dimensional space, a word is represented as a real-valued vector when it is embedded, and words with a similar meaning are positioned close together. Word embeddings are a learned, distributed, multi-dimensional representation of words, where each dimension represents a unique hidden characteristic of the word. Documents may be characterised as a collection of word vectors with a size equal to the document's word count by using the induced word representation provided by the embedding space (Butnaru et al. 2017).

Word2Vec uses real values from a vector space with a small number of dimensions to represent words as vectors. NLP researchers prefer Word2Vec's vector subtraction and addition features above those of other dimension reduction techniques like principal component analysis, latent semantic analysis, and latent dirichlet allocation (Shao et al. 2018). The limitations of Word2Vec in removing stop words should be considered. Beginning with the fact that Word2Vec overlooks the importance of each word in relation to the material being identified, correct categorization is difficult. Second, stop words can skew results since Word2Vec does not take into consideration how often words appear (Lilleberg et al. 2015).

### **2.3.3 Classification Algorithms**

Classification algorithms are a category of supervised learning techniques employed to predict the class label of unlabeled instances based on patterns learned from labeled training data. By analyzing a set of observations associated with known classes, a classification model learns to distinguish between categories such as *yes/no*,

*spam/not spam*, or different thematic or sentiment classes. In the context of text classification, these algorithms leverage linguistic and statistical features derived from textual content to assign documents, messages, or social media posts to predefined categories.

A variety of classification algorithms have been applied extensively in the literature, each offering distinct advantages in terms of interpretability, computational complexity, and adaptability to high-dimensional and imbalanced data. This section provides an overview of several popular classification algorithms frequently adopted for text classification tasks, highlighting their fundamental principles, historical usage, and relevant literature. Additional conceptual illustrations and comparisons of these algorithms can be found in the appendix (see Figure 17 and Figure 18 in Annex-I).

### **Popular Classification Algorithms**

**Naïve Bayes (NB):** Naïve Bayes is a probabilistic classifier grounded in Bayes' theorem, assuming conditional independence between features. Despite this simplifying assumption, NB has been widely and successfully employed in document categorization and social media analysis since its early adoption in the 1950s (Xing et al. 2002; Al-garadi et al. 2016). Its computational efficiency and robustness to sparse, high-dimensional text data make it a dependable baseline, effectively handling diverse data sources (e.g., raw text, network properties, and linguistic features).

**K-Nearest Neighbors (KNN):** K-Nearest Neighbors classifies instances based on the labels of their closest neighbors, using distance metrics to measure similarity. Its simplicity, interpretability, and non-parametric nature have led to widespread usage in text classification tasks (Shengyi Jiang et al. 2012). In multi-class scenarios, KNN assigns a label by considering the majority vote of the nearest neighbors. Although class imbalance can influence these votes, variants of KNN incorporating weighted neighbors or data-level adjustments can mitigate such biases (Talpur et al. 2020).

**Decision Trees (DT):** Decision Trees are rule-based classifiers that segment the feature space into regions defined by decision rules. They handle both discrete and continuous variables and can cope with missing values (Quinlan 1993). Their interpretability and direct mapping of decision criteria to human-readable structures have cemented their status as a foundational algorithm in machine learning. DTs are often used as building blocks in more complex ensemble methods and serve as valuable baselines in text classification.

**Random Forest (RF):** Random Forest is an ensemble method that aggregates multiple decision trees to improve accuracy and reduce overfitting. RF is frequently cited for its effectiveness in imbalanced scenarios (S. More et al. 2018), making it a strong candidate for text classification tasks where certain classes may be under-represented. Modifications such as cost-sensitive transformations and class-weight adjustments can enhance RF’s sensitivity to minority classes (C. Chen et al. 2004). The literature has consistently demonstrated that RF can yield superior classification accuracy across various domains, including textual data (Banerjee et al. 2018).

**Support Vector Machine (SVM):** Support Vector Machines aim to find an optimal hyperplane (or set of hyperplanes) to separate classes in the feature space. Their ability to handle high-dimensional data and flexibility through kernel functions (e.g., linear, polynomial, RBF) makes them well-suited for nuanced text representations (Akbani et al. 2004). SVMs are widely regarded as robust classifiers that demonstrate encouraging results in imbalanced settings (Wah et al. 2016). Multi-class extensions, such as one-against-one or one-against-all schemes, enable SVMs to classify data into more than two categories effectively (Yi Liu et al. 2005).

## Summary

Each of these popular classification algorithms—NB, KNN, DT, RF, and SVM—has a track record of usage in text classification tasks. Their differences in learning assumptions, handling of high-dimensional data, and sensitivity to class imbalances

offer a broad perspective on how various modeling strategies can be combined with advanced feature engineering approaches. Understanding their fundamental concepts and historical applications provides a strong foundation for evaluating their performance and suitability in specific text classification scenarios, including those addressed by the STREAM methodology.

### **2.3.4 Data and Algorithm Level Approaches**

Data-level approaches and algorithm-level approaches are the two broad groups of solutions to the class imbalance problem. The two broad classes can be subdivided even further into smaller classes (A. Ali et al. 2013). One can further categorise Data-Level techniques by either sampling data or choosing features. Depending on whether the sampled data was chosen at random or in accordance with a specified formula or algorithm, data sampling techniques can be further classified into over-sampling methods and under-sampling methods (T. M. Khoshgoftaar et al. 2007; Van Hulse et al. 2007). Over-sampling is the practise of replicating cases from the minority class into the provided dataset, either at random or with the use of a competent algorithm. Contrarily, under-sampling means consciously excluding samples from a collection of the provided dataset. Feature selection techniques are frequently used to improve classification performance alone, but they can also be used to pinpoint the most significant features (or traits) that can produce distinctive data for inter-class discrimination, even though they may lessen the effect of imbalanced datasets on classification accuracy (Malhotra 2015; H. Wang et al. 2015). Furthermore, Table 2.1 summarizes the studies that have focused on preprocessing techniques for text classification.

| Authors                                  | Datasets                                           | Classifiers/Algorithms                    | Preprocessing Methods                                              | Class Imbalance |
|------------------------------------------|----------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------|-----------------|
| (Song et al. 2005)                       | 20 Newsgroups and Reuters-21578                    | Linear SVM                                | Word stemming, stop-word removal, TF-IDF, and normalization        | No              |
| (Méndez et al. 2005)                     | Email corpora                                      | NB, SVM, and Adaboost                     | Tokenization, stop-word removal, and stemming                      | No              |
| (Toman et al. 2006)                      | English (Reuters Corpus) and Czech documents       | Multinomial NB                            | Word normalization, stop-word removal, stemming, and lemmatization | No              |
| (Pomikalek et al. 2007)                  | Reuters-21578 ModApte, 20 Newsgroups, and Springer | SVM, NB, KNN, Neural Networks, C4.5, etc. | Stop-word removal, stemming, and tokenization                      | No              |
| (A. Sun et al. 2009)                     | Reuters-21578, 20-Newsgroups, and WebKB            | SVM                                       | HTML tag removal, stop-word removal, and stemming                  | No              |
| (Y. Liu et al. 2009)                     | MCV1 and Reuters-21578                             | NB and SVM                                | Tokenization, stop-word removal, punctuation removal, and stemming | No              |
| (Goncalves et al. 2010)                  | MEDLINE                                            | SMO, RF, IBk (KNN), BayesNet, j48, etc.   | Pruning, stemming, and WordNet integration                         | No              |
| (Srividhya et al. 2010)                  | Reuters-21578                                      | N/A                                       | Stop-word removal, stemming, and TF-IDF                            | No              |
| (Clark et al. 2011)                      | Twitter                                            | N/A                                       | Fixing errors, abbreviations, slang, punctuation, etc.             | No              |
| (Munková et al. 2013)                    | N/A                                                | N/A                                       | Stop-word removal                                                  | No              |
| (Haddi et al. 2013)                      | Two movie review datasets                          | SVM                                       | Filtering, stemming, handling negations, and HTML cleanup          | No              |
| (Uysal et al. 2014)                      | Enron, Reuters-21578, and others                   | SVM                                       | Tokenization, stop-word removal, lowercasing, and stemming         | Yes             |
| (Duwairi et al. 2014)                    | Politics and movie datasets                        | NB, KNN, and SVM                          | Stop-word removal and stemming                                     | No              |
| (Ghag et al. 2015)                       | Movie documents                                    | Various sentiment classifiers             | Stop-word removal                                                  | No              |
| (Jianqiang 2015)                         | Twitter                                            | LR, NB, RF, and SVM                       | URL removal, stop-word removal, and acronym expansion              | No              |
| (Ayedh et al. 2016)                      | Various topics (politics, business, etc.)          | NB, KNN, SVM                              | Stop-word removal, stemming, and normalization                     | No              |
| (Kanan et al. 2016)                      | Newspaper articles                                 | NB, RF, and SVM                           | Stemming                                                           | No              |
| (Krouska et al. 2016)                    | Twitter                                            | C4.5, NB, KNN, and SVM                    | Stop-word removal, TF-IDF, and tokenization                        | No              |
| (Jianqiang and Xiaolin 2017)             | Twitter                                            | LR, NB, RF, and SVM                       | Stop-word removal, URL removal, and acronym expansion              | No              |
| (P. Sharma et al. 2017)                  | Twitter                                            | N/A                                       | Stop-word removal, URL removal, and stemming                       | No              |
| (HaCohen-Kerner, Rosenfeld, et al. 2018) | CLPsych datasets (2015, 2016, 2017)                | BN, SMO, and RF                           | Case modification, HTML tag removal, spelling correction, etc.     | No              |
| (Grootendorst et al. 2020)               | 20 Newsgroups, Reuters R8, and WebKB               | Linear SVM                                | Stop-word removal and stemming                                     | No              |
| (HaCohen-Kerner, Miller, et al. 2020)    | Various online databases                           | BN, SMO, and RF                           | HTML tag removal, punctuation removal, and spelling correction     | No              |
| (Pittaras et al. 2021))                  | Reuters-21578 and 20 Newsgroups                    | DNN                                       | Stop-word removal and tokenization                                 | No              |

Table 2.1: Studies that have dealt with text classification preprocessing

Approaches at the algorithmic level can be further divided into two categories: cost-sensitive approaches and hybrid/ensemble methods. In the former, misclassification suffers more severely when occurrences or trainers are given greater weight. A negative outcome prediction could be punished more harshly than a positive result forecast, for example, if the class of attention is positive. The classification output is the result of combining many classifiers built on the dataset, and popular group learning models that may be utilised as cost-sensitive strategies include Bagging and Boosting (Graczyk et al. 2010; Seiffert et al. 2010). The input dataset is divided into several training sets during bagging, and each training set is used to train a different classifier, therefore reducing the prediction variance. A technique called "boosting" combines the output of several classifiers using a weighted average of their misclassifications to produce a single final classification. This method, like "enhancing," employs various training data from the supplied dataset.

It is sometimes required to integrate strategies from the Data-Level and the Algorithmic-Level categories in order to address the class imbalance problem. Researchers have created ensemble learning techniques that use numerous basic classifiers to boost performance on imbalanced datasets. Additionally, there are a number of hybrid models in the literature that perform better at categorising imbalanced data by combining ensemble learning with data-level techniques (Majumder et al. 2020). The authors have addressed the issue of class imbalance through the strategic application of sampling, feature selection, and cost-aware methodologies. Therefore, the next section will focus on explaining these methods in detail.

### **2.3.5 Sampling Techniques**

It is challenging to create a reliable classifier from an imbalanced dataset since traditional classifiers frequently favour the majority class in order to maximise overall prediction accuracy. It is crucial to take into account the kind of sampling technique that should be employed in such circumstances. Although there are several sampling techniques available to handle data imbalance, there isn't a universally appli-

cable answer. The dataset's characteristics and the main learning objective influence the choice of sampling technique. Results from a study (Banerjee et al. 2018) indicate that the choice of sampling technique can significantly affect the difference in sensitivity and specificity of a trained model when compared to a non-sampling technique. In an imbalanced dataset, sampling is used to rebalance the sample space, reducing the impact of a skewed class distribution on the learning process. According to (López et al. 2013) sampling techniques' adaptability results from their independence of the chosen classifier.

### **Over-sampling**

By generating fresh minority class samples, over-sampling techniques have been found to be successful in addressing the issue of imbalanced datasets (Chawla et al. 2002). Randomly replicating minority samples and SMOTE are two frequently employed techniques for producing synthetic minority sample (Fernandez et al. 2018). By interpolating between minority samples and their closest neighbours in the feature space, SMOTE has been shown to be particularly successful at creating informative synthetic samples (Chawla et al. 2002; Fernandez et al. 2018). Teh et al. (2020) used over-sampling techniques to balance the classes in an effort to enhance classification performance. They evaluated four different classifiers, including KNN, SVM, multilayer perceptron, and DT, along with 73 over-sampling strategies. The researchers improved classification in datasets that were noticeably sparser and clustered by using imbalanced learning over-sampling approaches. After conducting their analysis (Teh et al. 2020), authors reported on the best over-sampling and classifier combinations. They discovered that using over-sampling techniques regularly outperformed no over-sampling techniques, hence improving the classification result over samplers. SVM was the best-performing classifier out of the four tested during the study. SVM was robust, accurate, and simpler to train on smaller datasets, according to the researchers. SVM was a versatile and useful option for classification challenges because of its capacity to produce nonlinear de-

cision boundaries using techniques intended for linear classifiers.

### **Under-sampling**

Under-sampling techniques are used to exclude samples from the dominant class in order to balance the distribution, which addresses the issue of an imbalanced class distribution. This approach has been found to be effective in improving the overall classification performance. One widely used under-sampling technique is Random Under-Sampling, which randomly eliminates examples from the majority class (M. A. Tahir et al. 2009). But it should be emphasised that Random Under-Sampling could lead to the loss of essential information in the majority class, which might have an adverse impact on classification performance. In light of this, it is crucial to carefully weigh the trade-off between the degree of under-sampling and its effect on classification performance. Other under-sampling methods, such as Tomek links and edited nearest neighbors, have also been proposed and shown to be effective in certain situations (López et al. 2013).

### **Blended approaches**

Hybrid approaches, as opposed to conventional statistical methods, integrate the finest aspects of both over-and under-sampling. These methods combine the two sampling strategies (Haixiang et al. 2017). It is crucial to emphasise that any available method may be used to re-sample to any desired ratio and that maintaining a precise balance between the population of the majority and minority groups is neither necessary nor sufficient. Numerous minority class cases were examined by (Napierala et al. 2015) along with how they affected learning classifiers from imbalanced data. Several publications attempted to find a way to automatically determine the optimal sample rate for a variety of issue situations and Imbalanced Ratios (Lu et al. 2016; Moreo et al. 2016). Several authors, including (Loyola-Gonzalez et al. 2019; Napierala et al. 2015), have conducted research on the effectiveness of various resampling techniques. The following are some of the key takeaways from these

papers:

1) It was demonstrated that an under-sampling technique saved much more time than an over-sampling strategy when hundreds of observations from a minority group are included in the dataset. 2) It was shown that when there are just a few dozen samples of a minority group, the over-sampling technique known as SMOTE is the way to utilise. 3) It is advisable to employ an alternate technique that combines SMOTE with under-sampling if the size of the training sample is too large. 4) When it comes to identifying outliers, SMOTE is somewhat more effective (Haixiang et al. 2017).

### **Advanced Synthetic Data Generation**

In addition to traditional over-sampling methods, advanced techniques like Generative Adversarial Networks have been employed to generate synthetic data for minority classes. Mariani et al. (2018) proposed Balancing GAN (BAGAN), which balances class distributions by generating synthetic images for minority classes. While this approach is primarily image-based, its underlying principles have inspired similar techniques in text classification. For instance, (Z. Liu et al. 2020) developed CatGAN, a GAN-based framework for generating synthetic text data to balance class distributions. Their experiments demonstrated that CatGAN could effectively generate realistic and diverse text samples.

### **Variational Autoencoders (VAEs)**

Variational Autoencoders have also been explored for synthetic data generation in imbalanced datasets. (Y.-D. Lin et al. 2022) introduced VAEs, which have been adapted to create synthetic data for minority classes in text classification. Variational Autoencoders generate new samples by learning the underlying data distribution and sampling from this distribution to create new data points. X. Du et al. (2020) applied Variational Autoencoders to generate synthetic minority class texts, showing improved classification performance in imbalanced text datasets.

## **Data Augmentation**

Data augmentation techniques, commonly used in image processing, have been adapted for text data to address class imbalance. Wei et al. (2019) introduced Easy Data Augmentation techniques for boosting performance on text classification tasks. Easy Data Augmentation involves simple operations such as synonym replacement, random insertion, random swap, and random deletion to generate augmented text samples. Their study showed that Easy Data Augmentation could significantly improve classifier performance on imbalanced datasets by increasing the diversity of training data.

## **Hybrid Approaches**

Hybrid approaches combining over-sampling and data augmentation have shown promise in recent studies. For example, Y. Chen et al. (2021) proposed a hybrid method that integrates data augmentation with SMOTE to generate a more diverse set of synthetic samples for minority classes. Their results indicated that this combination could effectively balance class distributions and enhance classifier performance on imbalanced text datasets.

## **Contextual Data Augmentation**

Recent advancements in natural language processing, particularly the use of pre-trained language models like BERT (Devlin et al. 2019), have enabled more sophisticated data augmentation techniques. X. Wu et al. (2019) introduced contextual augmentation methods that utilize BERT to generate contextually relevant synthetic samples for minority classes. This approach preserves the semantic coherence of the text while increasing the training data's diversity, leading to improved performance on imbalanced text classification tasks.

## **Threshold Moving**

Threshold moving is an algorithm-level technique designed to address class imbalance in binary classification by adjusting the decision threshold of classifiers. X. Zhang et al. (2021) proposed a simple threshold moving method and its extension using the bagging technique combined with binary relevance in multi-label classification. Their experiments on scene and medical datasets showed that applying threshold moving techniques significantly improved the performance of binary relevance based on a probabilistic classifier. For medical data, binary relevance with moved thresholds outperformed the classifier chain approach. The study found more noticeable improvements as the level of imbalance increased. Threshold moving techniques can be combined with any binary model-based approach, such as classifier chains, offering the potential for further performance improvements in imbalanced datasets.

## **Meta-learning Approaches**

Meta-learning, often referred to as "learning to learn," is an approach in machine learning where models are trained to adapt quickly to new tasks by leveraging prior experience (Finn et al. 2017). It aims to create models that can generalize from a small amount of data or from related tasks. Meta-learning has been employed to address class imbalance by training models to adapt to different levels of class imbalance. (Shu et al. 2019) introduced a meta-learning approach that learns a re-weighting function to dynamically adjust the importance of samples during training.

## **AutoML for Imbalance Learn**

Automated Machine Learning frameworks have been developed to automatically select and optimize models for imbalanced datasets. Auto-sklearn Feurer et al. (2019) include built-in methods to handle class imbalance, such as automated resampling, cost-sensitive learning, and ensemble techniques. These frameworks make it

easier for practitioners to develop robust models for imbalanced data without extensive manual tuning.

### **2.3.6 Synthetic Minority Over-sampling Technique**

By over-representing the under-represented minority, SMOTE may be utilised to solve the issue of a big majority in the training set. A statistically insignificant minority must make up at least one of a dataset's classes in order for it to be deemed imbalanced. When the datasets in question are not uniformly distributed, over-sampling techniques like SMOTE can be used. When a model is evaluated and applied to an imbalanced dataset, predictions might differ greatly. By applying SMOTE to an unequal dataset, the minority group will expand until it is the same size as the majority group (Amirneni 2019). The heuristic SMOTE is an over-sampling algorithm and by interpolating across closely spaced real-world examples, it creates synthetic samples from the underrepresented class. It is one of the most popular data collection methods now in use and has influenced the development of several over-sampling algorithms (Basgall et al. 2019; Fernandez et al. 2018). The SMOTE preparation method is now the "de facto" accepted method for learning from asymmetrical data. This can be attributed to the straightforward nature of the procedure's design as well as its adaptability to a wide variety of challenges when it is put into practise. Since its initial publication in 2002, SMOTE has been successfully implemented in a wide range of contexts. SMOTE has influenced several strategies for tackling the issue of class imbalance in addition to new supervised learning paradigms such as incremental learning, multilabel classification, semi-supervised learning, and multi-instance learning. It is commonly used to gauge how effectively an algorithm can adapt to skewed data. It is integrated into many software applications, both free and premium (Basgall et al. 2019; Fernandez et al. 2018). Empirical studies demonstrate that cost-sensitive learning performs well for two-class problems but poorly for challenges involving multi-class. It can be more difficult to include the multi-class cost function into any learning algorithms since there are

more consequences for a misclassified sample in multi-class tasks than there are in two-class tasks (X.-y. Liu et al. 2006). In contrast to random over sampling, which has the risk of over-fitting, the SMOTE generates new examples by interpolating existing instances (Chawla et al. 2002).

### **2.3.7 Feature Selection**

To train reliable prediction models, it is necessary to narrow down a large pool of potential characteristics to a manageable subset. Only a few possible advantages of variable and feature selection include improving prediction performance by overcoming the curse of dimensionality, measuring and storing requirements, training and usage periods, and data visualisation and understanding (Guyon et al. 2003; N. Kwak et al. 2002). When gathering data, application developers sometimes overlook the importance of certain elements, but they nevertheless take everything into consideration. Such techniques frequently provide a set of data with a number of properties that are either insignificant or just tenuously related. Numerous studies demonstrate how too many extraneous characteristics can muddle datasets and lead to misclassification of target samples. A wide range of features makes it difficult for domain specialists to extract information from categorization models that is relevant. As a result, several studies have been carried out to discover the most effective techniques for choosing which components to include and which ones to exclude in a classification model (Kamal et al. 2010).

In data mining, choosing features is an important step in the initial preparation of raw data. There are several feature selection techniques for two-class skewed data, and feature selection can be advantageous for datasets that are imbalanced. However, many real-world applications contain numerous classes of imbalanced data in addition to only two classes. Much progress has been made in classifying multi-class imbalanced datasets, but not in selecting features to exploit (L.-m. Du et al. 2015). The variables or characteristics you decide to use may be quite important when working with imbalanced data. Over-fitting the data to the model and intro-

ducing noise into the results are two ways in which having too many variables can hurt its generalizability and effectiveness. Variable selection procedures often entail removing unnecessary or redundant variables and choosing a smaller collection of variables to improve the model's performance (Ramos-López et al. 2021).

The practice of using feature selection to solve imbalanced datasets has grown in popularity within the data mining and machine learning communities in recent years. The strategies (resampling, etc.) outlined above for correcting data's unequal class distribution are focused on training data sampling (Moayedikia et al. 2017). A feature technique like feature selection employs a different approach to the imbalanced class problem than under- or over-sampling the training data to address the issue. The main goal is to determine the key traits that best represent both classes while also producing a collection of features that, ideally, closes the gap between classes in the set of data. Wrapper methods, filter methods, and embedding methods are the three broad categories of feature selection strategies (Abdulrauf Sharifai et al. 2020). Although filters may typically find a useful subset, this subset may not always be the greatest match for a particular classifier. It was therefore suggested to employ wrapper and embedding techniques to create a feature subset with more precise boundaries (Maldonado et al. 2018; P. Yang et al. 2013).

Mladenec et al. (1999) and Zheng et al. (2004) investigated the problem of identifying features for text classification using imbalanced data. Using feature selection methods such as Chi Square, Information Gain, and correlation coefficient (each method described below), both positive and negative features were selected independently before being explicitly integrated in their method. Different iterations of the odds ratio and the Information Gain measures are also presented with the express purpose of addressing class differences. NB and Regularized Logistic Regression were utilized as classifiers in their research, and the findings were promising for the proposed method. Yin et al. (2013) used Bayesian learning to solve the problem of unequal classes. The basis of this strategy was based on the presumption that the samples from the majority class have a significant influence on the overall

feature selection process. The effect of bigger class samples on methods of feature selection based on F-measure and AUC-ROC has been mitigated by the proposed strategy. Experimental data over synthetic features showed that the suggested technique effectively addressed the high-dimensional data and imbalanced class data problems.

C. Zhang et al. (2017) introduced a novel variable selection and skewed dataset that employs F-measure as a performance metric for feature selection rather than classification accuracy. The structural support vector machine (SSVM) approach used the highest F-measure metric to prioritize and weighted features for skewed data. A balanced vector of symmetrical uncertainty and SSVM were used to create a novel feature ranking algorithm that kept the best features. Subsets of features were then selected using a harmony search technique. The feature subsets include both under-represented and over-represented groups. The feature subsets include both groups with under-representation and those with over-representation. Correcting imbalanced classification of a high-dimensional dataset using this method produced positive outcomes (binary) after testing six datasets.

Tanha et al. (2020) have discussed the most significant boosting ensemble techniques for multi-class imbalance where LogitBoost algorithm performs better than other similar algorithms. However, boosting techniques are unable to demonstrate acceptable prediction performance in multi-class big datasets with a huge, imbalanced ratio. Additionally, they concluded that oversampling is a workable method for addressing the data's asymmetry. The different assessment metrics were compared using algorithm performance on common datasets, and it was shown that the multi-class Matthews correlation coefficient is the best metric for multi-class imbalanced data.

In order to cope with high-dimensional and unequal biomedical datasets, (Abdulrauf Sharifai et al. 2020) created rCBR-BGOA. They used a feature selection algorithm to find the right subsets of attributes that define the minority and majority

class. The suggested rCBR-BGOA technique is more effective because it combines the rCBR filter strategy with the BGOA wrapper method to choose a smaller subset of pertinent features from both binary and multi-class imbalance datasets. Performance metrics such as G-mean and Area under the curve (AUC) indicate that rCBR-BGOA can improve classifier performance for high-dimensional and imbalanced datasets.

### **Cost-Sensitive Approach**

Beyond the inherent imbalance in class distributions, the severity of different types of misclassification can vary significantly (Elrahman et al. 2013). Cost-sensitive learning addresses this issue by assigning different costs to distinct misclassification errors, guiding the classifier to minimize not just overall error rates but also the more consequential misclassifications (He et al. 2009; Yanmin Sun et al. 2009; Iranmehr et al. 2019; Tang et al. 2014). For example, failing to detect a rare but critical class (e.g., a severe form of cyberbullying or fraudulent behavior) can be assigned a higher penalty than a more common, less harmful misclassification (Khan et al. 2017; Yu et al. 2019).

Cost-sensitive learning can be implemented directly by modifying the learning algorithm, integrating cost matrices, or by using wrapper techniques that adjust decision thresholds (Charles X Ling et al. 2008; P. D. Turney 1995; Drummond et al. 2000; Charles X. Ling, Q. Yang, et al. 2004). Although many cost-sensitive strategies were initially developed for binary imbalanced datasets, researchers have explored extending them to multi-class scenarios, which is substantially more complex (S. Wang and Yao 2012). Approaches like class decomposition (e.g., one-vs-all or one-vs-one) allow decomposing multi-class problems into binary tasks before applying cost-sensitive methods (Agrawal et al. 2015; Krawczyk 2016; Lango and Stefanowski 2018). Studies have also investigated optimizing cost-based parameters and evaluating multi-class imbalanced data under various cost-sensitive criteria (Elkan 2001; X.-y. Liu et al. 2006; Lozano et al. 2008).

In text classification, cost-sensitive techniques serve as useful baseline tools, especially when minority classes represent rare yet crucial categories, such as detecting rare instances of harmful content. By applying cost-sensitive adjustments—either through cost-sensitive learners or threshold modifications—researchers can improve classifier vigilance toward underrepresented classes without fundamentally altering data distributions.

### **Information Gain**

Information Gain is a frequently employed feature selection technique in text classification (Forman 2003; Quinlan 1993). It measures how much a feature reduces uncertainty about the class, ranking features according to their ability to discriminate among classes. Studies have shown Information Gain’s effectiveness in improving classifier accuracy, reducing dimensionality, and removing redundant or irrelevant features (Aggarwal et al. 2012; Lee et al. 2006; Pau et al. 2023).

Applied to text classification, Information Gain has demonstrated efficacy in various domains. For instance, (Toolan et al. 2010) improved spam email detection using Information Gain, while (Tan et al. 2008) found it beneficial for sentiment classification tasks. By focusing on the most informative terms, Information Gain helps ensure that classifiers concentrate on features genuinely relevant to class separation, potentially aiding performance even in imbalanced class conditions (Yeh et al. 2021).

### **Chi-Square**

The Chi-square test is another widely adopted statistical measure for feature selection in text classification. By examining the independence between terms and classes, Chi-square highlights terms that are strongly associated with certain classes (Forman 2003). Retaining features with high Chi-square scores can reduce the feature space size and complexity, leading to more efficient and accurate models.

As a baseline selection approach, Chi-square filtering is particularly useful when evaluating models that rely on classic Bag-of-Words or more advanced embedding-

based representations. Its ability to emphasize terms with solid class associations can aid in achieving better performance, especially in contexts where some classes are underrepresented and require stronger lexical cues for detection.

## **Correlation**

Correlation-based feature selection identifies the most relevant features by evaluating their correlation—or symmetrical uncertainty—with the class labels (Hall 2000). Features exhibiting stronger correlation with the class can be retained, while those deemed less informative or redundant are removed. This method has shown improvements in classifier accuracy and simplicity across various textual domains (Duangsoithong et al. 2010).

Correlation-based selection has proven advantageous in fields such as sentiment analysis (Al-Harbi 2019), ensuring that classifiers focus on terms genuinely reflecting class distinctions. Moreover, research suggests that correlation-based filtering can be effective in imbalanced scenarios, where focusing on strongly correlated features helps minority classes gain representation in the feature set (Alam et al. 2019; Gao et al. 2015). Thus, correlation-based selection, like the other techniques discussed, can serve as a valuable baseline for refining textual features before classification.

## **Summary**

The cost-sensitive approach, Information Gain, Chi-square, and correlation-based selection stand as core baseline methods in text classification research. Each technique contributes to addressing fundamental challenges—such as class imbalance, feature redundancy, and high dimensionality—by refining feature sets and guiding classifiers toward more meaningful distinctions. When combined with various text representations (from Bag-of-Words to Word2Vec), these methods offer practical and proven strategies for establishing robust baselines against which more sophisticated approaches can be compared.

### 2.3.8 Model Performance Evaluation

Evaluating model performance is a critical stage in developing a machine learning classifier. Traditional metrics such as overall accuracy, precision, and recall have long been considered standard for measuring effectiveness in binary classification problems. However, these measures often fail to provide meaningful insight when class distributions are skewed, as high performance on the majority class can mask poor recognition of minority classes (Teh et al. 2020).

Accuracy, in particular, can be misleading in imbalanced settings. Even with high accuracy, a classifier may struggle to correctly identify instances from rare classes. Consequently, research has moved beyond simple accuracy-based evaluation to metrics that reflect performance on underrepresented classes. Various performance measures have been proposed to better represent classifier behavior under class imbalance (S. Wang 2011; He et al. 2009; Yanmin Sun et al. 2009).

A fundamental tool for understanding classifier outcomes is the confusion matrix, which, in its simplest form, is defined for two classes (positive and negative). Table 2.2 provides an example structure of such a confusion matrix:

Table 2.2: Confusion Matrix

|                | Positive Prediction | Negative Prediction |
|----------------|---------------------|---------------------|
| Positive Class | True Positive (TP)  | False Negative (FN) |
| Negative Class | False Positive (FP) | True Negative (TN)  |

From the confusion matrix, a classifier's outcomes are categorized as:

- TP (True Positives): Correctly classified positive instances.
- TN (True Negatives): Correctly classified negative instances.
- FP (False Positives): Negative instances incorrectly labeled as positive.
- FN (False Negatives): Positive instances incorrectly labeled as negative.

In imbalanced data scenarios, the minority class is typically considered "posi-

tive” to focus on its detection quality. Performance metrics derived from the confusion matrix can be grouped into single-class measures, emphasizing a particular class (often the minority one), and overall performance measures, aiming for balanced recognition across classes (S. Wang 2011).

### Single-Class Performance Measures

Single-class performance measures are adapted from information retrieval metrics and include recall, precision, and the F-measure. These are defined as:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad \text{F-measure} = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

(with  $\beta = 1$  as a common default setting, giving equal weight to precision and recall.)

In the context of imbalanced classification, recall captures how many of the minority class instances are correctly identified, while precision indicates how many of the predicted positives are truly positive (He et al. 2009; Yanmin Sun et al. 2009). The F-measure combines both, reflecting the trade-off between them and often providing a more informative criterion than accuracy. This approach has shown superiority in highlighting performance where minority class detection is vital (S. Wang 2011).

### Overall Performance Measures

Other metrics attempt to provide a class-balanced view of performance. The G-mean, introduced by (Kubat and Matwin 1997), takes the geometric mean of recall values for both positive and negative classes. A high G-mean indicates that the classifier is performing well on all classes, not just the majority class.

The use of ROC (Receiver Operating Characteristic) curves and the Area Under

the ROC Curve (AUC) is another prominent method for evaluating classifiers in imbalanced domains (Bradley 1997; Charles X. Ling, J. Huang, et al. 2003; Fawcett 2006). The ROC curve plots the true positive rate against the false positive rate, illustrating the possible trade-offs between detection of positives and avoidance of false alarms. The AUC, measuring the area under this curve, provides a single scalar measure between 0 and 1. An AUC of 0.5 indicates random performance, while values closer to 1 suggest more effective discrimination (Hand et al. 2001). Though widely applied, ROC/AUC analysis is most directly suited for binary classification and does not inherently scale to multi-class scenarios (S. Wang 2011).

## **Kappa**

The Kappa statistic, originally devised to measure inter-rater agreement, has been adopted as a performance measure for classifiers (Cohen 1960; Vieira et al. 2010). Kappa adjusts observed accuracy by accounting for the agreement expected by chance, providing a normalized scale that helps distinguish genuine improvement from random labeling. Ranging from values below 0.00 (poor agreement) to values near 1.00 (almost perfect agreement) (Landis et al. 1977), a higher Kappa indicates that the classifier is performing meaningfully better than random guessing. Although Kappa handles chance agreement, it makes certain assumptions about data independence and other factors (McHugh 2012). Despite these constraints, it remains a valuable metric for assessing classifier effectiveness in skewed datasets, as it accounts for the underlying class distributions (M. A. U. H. Tahir et al. 2019).

## **Summary**

Model performance evaluation in imbalanced classification extends beyond simple accuracy. From single-class metrics like precision and recall, to balanced performance measures such as G-mean and AUC, and advanced statistics like Kappa, a range of metrics has been developed to capture nuanced classifier behavior in skewed data domains (Taghi M. Khoshgoftar et al. 2004; Kubat, Holte, et al. 1998;

Japkowicz 2008; Seliya et al. 2009; Y. Sun et al. 2006; Zhen et al. 2012). When dealing with multi-class imbalances, the appropriate selection and interpretation of these metrics become even more essential, ensuring that minority classes—though few in number—are accorded sufficient importance in model evaluation.

## 2.4 STREAM - Related work

Due to its high accuracy rate in determining the semantic similarity of words and relatively cheap computing cost, the adoption of Word2Vec, a word embedding approach created by (Mikolov, Sutskever, et al. 2013), has grown in popularity. Word embeddings may be easily learned from raw text using the BoW model or the Skip-gram model, and word connections can be captured using the built-in vector representation model. Word2Vec is effective in various fields, such as NLP (Suqi Jiang et al. 2016; Moran et al. 2016), sound recognition (Vijayakumar et al. 2017), and biology (Duong et al. 2017). Among these, NLP is a popular area where Word2Vec has been used for different tasks, such as story detection (Moran et al. 2016), sentiment analysis (Suqi Jiang et al. 2016), and text classification (Feng et al. 2016; Lilleberg et al. 2015)

Lilleberg et al. (2015) utilized Word2Vec to generate vector representations of documents and concatenated them with TF-IDF vectors as additional features to achieve improved classification accuracy in SVM. On the other hand, (Feng et al. 2016) proposed a Word2Vec-based method with NB classifier that used the conditional probability of a word based on its similarity with other words. However, their model's results were inconclusive due to the lack of evidence showing that the cumulative density functions could hold to one. Moreover, Rui et al. (2016) proposed a model that used Word2Vec as a tool to achieve unsupervised feature reduction by clustering similar words using the K-Means method (Ge et al. 2017). Even though Word2Vec's inherent process is not fully understood, it has proved successful in a number of different fields.

Fan et al. (2017) employed Word2Vec to train the Chinese word vector from an extensive corpus,. A multilabel classification technique was then used to categorize the questions after they generated the semantic feature vector of a Chinese question based on the training word vector. Similarly, to demonstrate which technique is best for an imbalanced dataset, Esmeli et al. (2020) explored with several class imbalance strategies using the Word2Vec method to calculate product similarities. However, Ashrafi et al. (2020) found that contextual embeddings, such as those used by the BERT model, are necessary to produce better results for imbalanced datasets. They also proposed a statistical approach that yielded better results compared to Focal Loss. For small sets with significant imbalance, Padurariu et al. (2019) discovered that basic text representations like BoW or TF-IDF perform better than more complicated embeddings like Glove or Doc2Vec.

Zeng et al. (2015) faced the challenge of class imbalance in their dataset of high-resolution in situ hybridization images of developing mouse brain. They used a deep convolutional neural network and a binary-classification technique to extract information from the images. Additionally, they built picture-level representations using the BoW technique using invariant image feature descriptors as a foundational representation. Similarly, Glazkova (2020) examined synthetic oversampling techniques for the challenge of multi-class topic classification by combining machine learning algorithms with several text representations (BoW, TF-IDF, Word2Vec). They discovered that using oversampling approaches may greatly boost the accuracy of classification, particularly when using KNN and SVM algorithms. S. Wu (2018) proposed a novel feature selection algorithm based on multi-granularity latent Dirichlet allocation and a cost-sensitive SVM to construct a fault diagnosis model for maintenance text. Their algorithm effectively improved the classification performance of minority classes while maintaining the accuracy of majority classes. Table 2.3 provides a list of relevant studies in the text classification domain and the nature of their class imbalance distribution.

## 2.5 Chapter Summary

The current research on class imbalance in text classification has explored various techniques to mitigate the issue, including the use of simple text representations and deep learning-based tools. Effective strategies for dealing with class imbalance in high-dimensional feature spaces, however, are still lacking in the literature. More precisely, methods that can effectively resolve class imbalance without needing a lot of processing power or training data are needed. This research gap offers a chance to create fresh approaches to class imbalance in text classification that can boost classification precision for imbalanced datasets. Additionally, given the prevalence of class imbalance in many real-world applications, such as medical diagnosis and fraud detection, addressing this issue will continue to be a critical area of focus for machine learning research. In the next chapter, the design of the proposed STREAM method is presented.

| Authors                            | Dataset                                                                                                 | Text Representation Techniques        | Classifiers                                        | Imbalance   |
|------------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------|-------------|
| (A. Sun et al. 2009)               | Reuters-21578, 20-Newsgroups, and WebKB                                                                 | TF-IDF                                | SVM                                                | Multi-class |
| (Y. Liu et al. 2009)               | MCV1 and Reuters-21578                                                                                  | TF-IDF                                | NB and SVM                                         | Multi-class |
| (Ah-Pine et al. 2016)              | Obama-McCain-Debate (OMD), Health Care Reform (HCR), and Imagiweb (IW)                                  | BoW                                   | LR and DT                                          | Binary      |
| (N. Sharma et al. 2016)            | DBWorld, Reuters-21578, and 20-Newsgroups                                                               | BoW                                   | Additive NB for Multinomial Classification (ANBMC) | Multi-class |
| (Moreo et al. 2016)                | Reuters-21578, OHSUMED-S, and RCV1-v2                                                                   | BoW                                   | Linear SVM                                         | Binary      |
| (Heap et al. 2017)                 | Reuters-21578, ICG DRC, and OHSUMED                                                                     | BoW                                   | Multinomial NB and SVM                             | Multi-class |
| (Carrillo-de-Albornoz et al. 2018) | eDiseases dataset ( <a href="https://zenodo.org/record/1479354">https://zenodo.org/record/1479354</a> ) | BoW, Word2Vec, and concept embeddings | NB, SMO, RF, and vote                              | Multi-class |
| (Moraes et al. 2018)               | Reviews from movies and about GPS devices, books, and cameras collected from Amazon                     | BoW and TF-IDF                        | ANN and SVM                                        | Binary      |
| (Rudkowsky et al. 2018)            | Parliament Austria, 2013                                                                                | BoW and word embedding                | Multinomial NB                                     | Multi-class |
| (Eklund 2018)                      | Reuters-21578                                                                                           | BoW, GloVe, and TF-IDF                | ANN and SVM                                        | Multi-class |
| (Nguyen et al. 2018)               | Amazon                                                                                                  | TF-IDF                                | LR, SVM, and XGBoost                               | Binary      |
| (Padurariu et al. 2019)            | Corpus of financial and accounting sectors                                                              | BoW and TF-IDF                        | DT, LR, and SVM                                    | Multi-class |
| (Grootendorst et al. 2020)         | 20-Newsgroups, Reuters R8, WebKB, and Cade12                                                            | TF-IDF, BoW, Word2Vec, and GloVe      | Linear SVM                                         | Multi-class |
| (Yan et al. 2020)                  | 20-Newsgroups, WebKB, Reuters 52, and Amazon                                                            | BoW                                   | KNN                                                | N/A         |
| (Pittaras et al. 2021)             | Reuters-21578 and 20-Newsgroups                                                                         | Word2Vec                              | DNN                                                | Multi-class |
| (D. Ali et al. 2021)               | Urdu Point, BBC Urdu, and Geo News Channel                                                              | BoW, TF-IDF, and word embedding       | DT, KNN, LR, Multinomial NB, RF, and SVM           | Multi-class |
| (Srinivasan et al. 2021)           | Code-mixed social text data                                                                             | TF-IDF                                | LR, NB, RF, SVM, and XGBoost                       | Multi-class |

Table 2.3: References to the dataset’s literature, as well as examples of text representation methods, classifiers, and imbalance



## 3 | **STREAM Methodology**

### 3.1 Introduction

The exponential growth of textual data in the digital age has necessitated advanced methodologies capable of extracting meaningful patterns from unstructured text. In the domain of NLP, text classification continues to be a challenging task, particularly in cases involving multi-class imbalanced datasets (W.-J. Lin et al. 2013). Conventional techniques frequently fail to capture nuanced semantic relationships and adequately represent minority classes, resulting in poor classification performance. To address these challenges, this research proposes the STREAM, an novel framework that utilizes statistical techniques and semantic analysis to enhance text classification.

STREAM is a comprehensive methodology that integrates SO with extensive feature engineering techniques. By employing foundational principles such as PMI, STREAM reveals latent semantic structures within large text corpora. The proposed approach produces semantically enriched feature representations that are sensitive to word distributions across different classes, thus enhancing classification performance in imbalanced datasets. The novelty of STREAM relies in its ability to combine statistical measures of word associations with SO relative to particular classes, providing a substantial advancement in the domain of NLP and machine learning.

Importantly, STREAM is designed as an interpretable and computationally efficient alternative or complement to increasingly popular neural embedding methods

(e.g., transformers like BERT). Although these neural models can interpret contextual nuance, they frequently require substantial computational resources and large labeled datasets. On the contrary, STREAM’s statistical underpinnings allow for a more transparent understanding of how particular words influence classification decisions, particularly in scenarios of class imbalance. An overview of the STREAM methodology, with both core and optional steps, is illustrated in Figure 3.1.

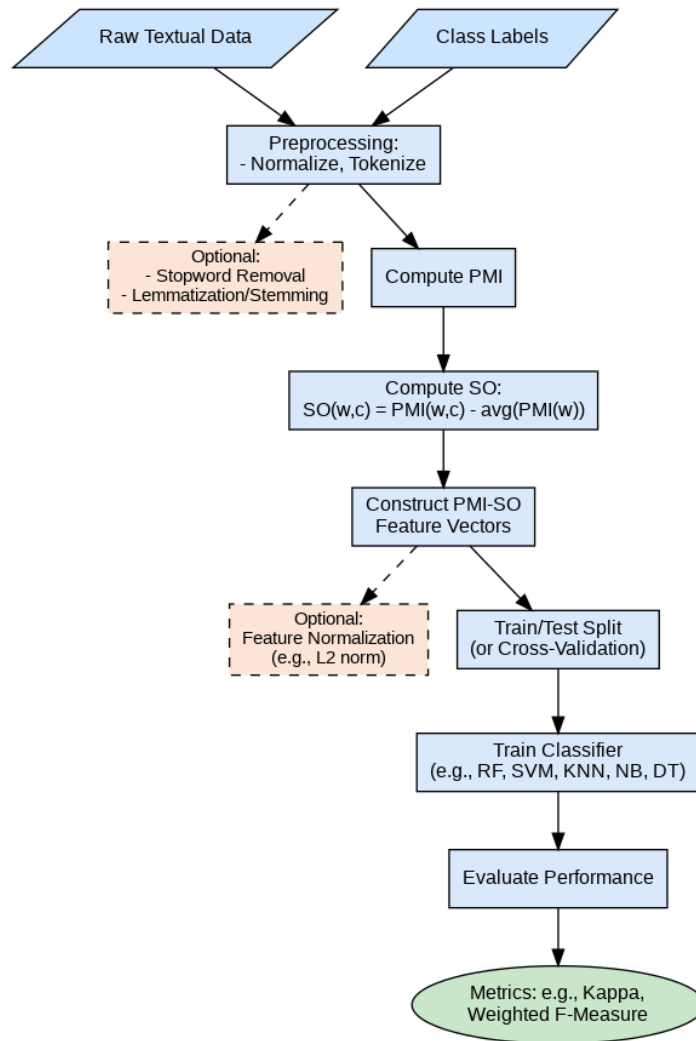


Figure 3.1: An overview of the STREAM methodology highlighting core and optional steps.

## 3.2 Understanding STREAM

STREAM signifies a paradigm shift in text categorization by focusing on the integration of SO with advanced feature engineering. Unlike conventional frequency-based

methods, STREAM utilizes statistical methods to capture deeper semantic relationships between words and their contextual usage. The key components of STREAM include the computation of PMI, estimation of SO, and the generation of document-level feature vectors using the combined PMI-SO approach.

At its core, STREAM creates a semantic space where words and documents are depicted as vectors informed by their class-specific associations. By calculating PMI to quantify word associations and estimating SO to capture the directional inclination of words towards particular classes, STREAM improves the classifier's ability to detect nuanced distinctions between classes. This is particularly crucial in multi-class classification tasks where class imbalance may result in underrepresentation of minority classes in the model's predictions.

Another significant advantage of the STREAM methodology is its inherent flexibility in integrating various class imbalance handling strategies. Whether by applying oversampling techniques (e.g., SMOTE), undersampling methods, or cost-sensitive adjustments, STREAM's pipeline can accommodate these approaches seamlessly. This adaptability ensures that researchers can tailor the class imbalance mitigation technique to the unique attributes of their dataset and classification goals, improving STREAM's overall robustness and applicability to various text classification scenarios.

The substantial impact of STREAM on NLP and machine learning research lies in its methodological innovation. It offers a comprehensive framework for generating semantically informed feature representations directly from the data, independent of external embeddings or frequency-based features like TF-IDF. This advancement has profound implications for numerous applications, including sentiment analysis, topic modeling, and other fields requiring nuanced semantic understanding.

## 3.3 Methodological Framework

The STREAM approach comprises a series of carefully designed processes that collectively contribute to generating robust and semantically informed feature vectors appropriate for classification tasks. This section provides an in-depth discussion of each component, highlighting the theoretical underpinnings, methodological decisions, and the justifications behind the choices made.

### 3.3.1 Data Acquisition and Preprocessing

The initial phase of STREAM focuses on collecting raw textual data along with their corresponding class labels, which serve as the primary inputs for the classification task. The specific datasets chosen for this comprehensive evaluation and the rationale behind their selection, are detailed in subsequent chapters. Ensuring that the data is appropriate for further analysis requires careful preprocessing, an critical step in transforming raw text into a standardized and analytically meaningful format.

Preprocessing begins with text normalization, a essential process that converts all textual data to a uniform standard. This involves converting all characters to lowercase, eliminating punctuation, numbers, and other extraneous symbols. By reducing variability and noise in the data, normalization lays the groundwork for more consistent and reliable feature extraction.

Another critical preprocessing step is tokenization, which divides the normalized text into individual units, usually words or terms. These tokens constitute the fundamental elements for developing the feature representations that will subsequently be enriched by PMI-SO transformations.

Further optional steps may further refine the data. For instance, the removal of stop words eliminates frequently used terms that contribute little to discriminating between classes, while lemmatization or stemming reduces words to their base

or root forms. implementing these optional procedures can improve the generalizability of the analysis by reducing the impact of minor morphological variations between words.

In summary, by systematically executing these preprocessing techniques—normalization, tokenization, and optionally stop words removal, lemmatization, or stemming—the raw input data is transformed into a cleaner, more coherent form. This guarantees that further phases of the STREAM methodology can effectively uncover latent semantic structures and ultimately improve classification performance.

### 3.3.2 Feature Engineering Using PMI and SO

Feature engineering refers to the process of generating or extracting features from raw data or corpora, as well as the development of novel features derived from existing features (Pattnaik et al. 2018). Feature engineering is the crux of the STREAM methodology, converting raw textual data into meaningful representations that machine learning algorithms can efficiently employ (Mehta 2017). The focus is on capturing semantic and statistical attributes of the text that are directly relevant to the classification task through the integrated application of PMI and SO.

#### Pointwise Mutual Information (PMI)

PMI is a statistical measure employed to assess the association strength between two words by comparing the probability of their co-occurrence with the probabilities of their individual occurrences. It is defined as:

$$\text{PMI}(w_i, w_j) = \log_2 \left( \frac{p(w_i, w_j)}{p(w_i) \cdot p(w_j)} \right) \quad (1)$$

Where:

- $p(w_i, w_j)$  is the probability of words  $w_i$  and  $w_j$  occurring together within a specified context window.

- $p(w_i)$  and  $p(w_j)$  are the marginal probabilities of  $w_i$  and  $w_j$ , respectively.

PMI effectively measures how much more likely two words co-occur than if they were independent (Peter D. Turney et al. 2003). High PMI values signify a robust correlation between words, which can be indicative of semantic or contextual relationships. The PMI method devised by Turney does not depend on predetermined semantic criteria, enabling users to easily apply the technique in many situations (Garrett et al. 2018).

### Semantic Orientation (SO)

SO enhances PMI by integrating class-specific information, assessing the correlation of a word with a particular class. It captures the directional inclination of words towards certain classes, which is very useful in classification tasks involving imbalanced datasets.

For each word  $w$  and class  $c$ , SO is calculated as:

$$SO(w, c) = PMI(w, c) - PMI(w, \bar{c}) \quad (2)$$

Where:

- $PMI(w, c)$  is the PMI between word  $w$  and class  $c$ .
- $PMI(w, \bar{c})$  is the average PMI between word  $w$  and all other classes except  $c$ .

A positive SO value indicates that the word is more strongly associated with class  $c$  than with other classes, while a negative value suggests the opposite. By calculating SO for all words across all classes, a feature space is created that captures the semantic orientation of words relative to each class.

### Novelty of PMI-SO for Feature Engineering

**Theoretical Foundations of PMI and SO** PMI is a measure from information theory that quantifies the relationship between two discrete events, such as a word and

a class label. It compares the probability of observing the two events together (joint probability) with the probabilities of observing them independently (marginal probabilities). For a word  $w$  and class  $c$ , PMI is defined as:

$$\text{PMI}(w, c) = \log_2 \left( \frac{p(w, c)}{p(w) \cdot p(c)} \right) \quad (3)$$

Where:

- $p(w, c)$  is the joint probability of word  $w$  and class  $c$ .
- $p(w)$  is the probability of word  $w$  occurring in the corpus.
- $p(c)$  is the probability of class  $c$ .

PMI quantifies how much more (or less) likely the word and class co-occur than if they were independent. A high PMI indicates a strong association, which is valuable for feature engineering as it identifies informative words for classification.

SO extends PMI by adjusting it to focus on the directional association of a word towards a specific class compared to other classes. For a word  $w$  and class  $c$ , SO is defined as:

$$\text{SO}(w, c) = \text{PMI}(w, c) - \overline{\text{PMI}}(w) \quad (4)$$

Where:

- $\overline{\text{PMI}}(w) = \frac{1}{|C|} \sum_{c' \in C} \text{PMI}(w, c')$  is the average PMI of word  $w$  across all classes.

SO quantifies the strength of a word's association with a certain class in relation to all classes. It augments the feature space by prioritizing words that can effectively discriminate between classes, which is essential in multi-class classification. The integration of PMI and SO produces a feature space that explicitly represents the inclination of words towards particular classifications. In contrast to external embeddings (e.g., Word2Vec, BERT), which learn general semantic representations, the

PMI-SO approach is essentially class-sensitive and transparent. This interpretability allows researchers and practitioners to understand which words drive classification decisions and how these decisions might be affected by skewed class distributions.

The PMI-SO for each class instance can be calculated in Python Natural Language Toolkit (NLTK) library<sup>1</sup> or using Weka’s AffectiveTweets library (Bravo-Marquez et al. 2019), or a combination of both Python and Weka libraries.

**Effectiveness for Imbalanced Multi-Class Classification** In imbalanced datasets, minority classes have fewer instances, resulting in fewer occurrences of associated words. Traditional frequency-based features may underrepresent these classes. PMI automatically assigns elevated ratings to word-class pairs that co-occur more frequently than predicted, even if the absolute frequency is low. This amplifies the significance of words in minority classes.

SO further enhances words uniquely linked with minority classes by modifying for the average PMI across all classes. This increases the sensitivity of features to minority class characteristics, hence enhancing the classifier’s ability to identify and accurately classify instances from these classes.

**Novelty and Contribution** This research presents a novel contribution through the combined application of PMI and SO for feature engineering in multi-class imbalanced text classification. It leverages statistical measures to improve feature representation, providing an interpretable, statistically based alternative to neural embeddings. This approach offers transparency in how features contribute to classification, which is valuable for understanding and improving models.

### **Detailed Calculations with Numerical Examples**

To illustrate how PMI and SO work in practice and how they help mitigate class imbalance, consider a simplified example with a small corpus of documents belong-

---

<sup>1</sup><https://www.nltk.org/>

ing to three classes:  $C = \{\text{Class 1}, \text{Class 2}, \text{Class 3}\}$ , with Class 3 being the minority class.

### Corpus Overview

- **Total Documents** ( $|D| = 10$ )
- **Class Distribution:**
  - **Class 1:** 5 documents
  - **Class 2:** 4 documents
  - **Class 3:** 1 document

Table 3.1: Word Occurrences in Classes

| Word  | Class 1 | Class 2 | Class 3 | Total Frequency $f(w)$ |
|-------|---------|---------|---------|------------------------|
| $w_1$ | 10      | 2       | 0       | 12                     |
| $w_2$ | 5       | 15      | 1       | 21                     |
| $w_3$ | 0       | 0       | 5       | 5                      |
| $w_4$ | 3       | 0       | 0       | 3                      |
| $w_5$ | 2       | 2       | 0       | 4                      |
| $w_6$ | 0       | 0       | 2       | 2                      |

**Vocabulary and Word Occurrences** Total word occurrences ( $N = 47$ ).

**Calculating Marginal Probabilities** For each word  $w$ :

$$p(w) = \frac{f(w)}{N}$$

Compute  $p(w)$  for each word:

- $p(w_1) = \frac{12}{47} \approx 0.255$
- $p(w_2) = \frac{21}{47} \approx 0.447$
- $p(w_3) = \frac{5}{47} \approx 0.106$

- $p(w_4) = \frac{3}{47} \approx 0.064$
- $p(w_5) = \frac{4}{47} \approx 0.085$
- $p(w_6) = \frac{2}{47} \approx 0.043$

Marginal probabilities for each class  $c$ :

$$p(c) = \frac{f(c)}{|D|}$$

Assuming  $f(c)$  is proportional to the number of documents in class  $c$ :

- $p(\text{Class 1}) = \frac{5}{10} = 0.5$
- $p(\text{Class 2}) = \frac{4}{10} = 0.4$
- $p(\text{Class 3}) = \frac{1}{10} = 0.1$

**Calculating Joint Probabilities** For each word  $w$  and class  $c$ :

$$p(w, c) = \frac{f(w, c)}{N}$$

Compute  $p(w, c)$  for selected word-class pairs:

- $p(w_1, \text{Class 1}) = \frac{10}{47} \approx 0.213$
- $p(w_1, \text{Class 2}) = \frac{2}{47} \approx 0.043$
- $p(w_1, \text{Class 3}) = 0$
- $p(w_2, \text{Class 1}) = \frac{5}{47} \approx 0.106$
- $p(w_2, \text{Class 2}) = \frac{15}{47} \approx 0.319$
- $p(w_2, \text{Class 3}) = \frac{1}{47} \approx 0.021$
- $p(w_3, \text{Class 3}) = \frac{5}{47} \approx 0.106$
- $p(w_6, \text{Class 3}) = \frac{2}{47} \approx 0.043$

**Calculating PMI Values** Using the formula:

$$\text{PMI}(w, c) = \log_2 \left( \frac{p(w, c)}{p(w) \cdot p(c)} \right)$$

**Example 1: PMI for  $w_1$  and Class 1**

- $p(w_1, \text{Class 1}) \approx 0.213$
- $p(w_1) \approx 0.255$
- $p(\text{Class 1}) = 0.5$

$$\text{PMI}(w_1, \text{Class 1}) = \log_2 \left( \frac{0.213}{0.255 \times 0.5} \right) = \log_2 \left( \frac{0.213}{0.1275} \right) = \log_2(1.6706) \approx 0.737$$

**Example 2: PMI for  $w_2$  and Class 2**

- $p(w_2, \text{Class 2}) \approx 0.319$
- $p(w_2) \approx 0.447$
- $p(\text{Class 2}) = 0.4$

$$\text{PMI}(w_2, \text{Class 2}) = \log_2 \left( \frac{0.319}{0.447 \times 0.4} \right) = \log_2 \left( \frac{0.319}{0.1788} \right) = \log_2(1.783) \approx 0.832$$

**Example 3: PMI for  $w_3$  and Class 3 (Minority Class)**

- $p(w_3, \text{Class 3}) \approx 0.106$
- $p(w_3) \approx 0.106$
- $p(\text{Class 3}) = 0.1$

$$\text{PMI}(w_3, \text{Class 3}) = \log_2 \left( \frac{0.106}{0.106 \times 0.1} \right) = \log_2 \left( \frac{0.106}{0.0106} \right) = \log_2(10) \approx 3.322$$

**Observation:**

The PMI for  $w_3$  and Class 3 is significantly higher than the PMIs for other word-class pairs, even though  $p(\text{Class 3})$  is small. This demonstrates how PMI amplifies the association between  $w_3$  and the minority class.

**Computing Average PMI for Each Word** Compute the average PMI for each word:

$$\overline{\text{PMI}}(w) = \frac{1}{|C|} \sum_{c' \in C} \text{PMI}(w, c')$$

Assuming that PMI is zero when  $p(w, c) = 0$ .

**For  $w_3$ :**

- $\text{PMI}(w_3, \text{Class 1}) = 0$
- $\text{PMI}(w_3, \text{Class 2}) = 0$
- $\text{PMI}(w_3, \text{Class 3}) \approx 3.322$

Compute average PMI:

$$\overline{\text{PMI}}(w_3) = \frac{1}{3}(0 + 0 + 3.322) = 1.107$$

**Computing SO Values** Compute SO for each word-class pair:

$$\text{SO}(w, c) = \text{PMI}(w, c) - \overline{\text{PMI}}(w)$$

**For  $w_3$  and Class 3:**

$$\text{SO}(w_3, \text{Class 3}) = 3.322 - 1.107 = 2.215$$

**For  $w_3$  and Classes 1 and 2:**

$$\text{SO}(w_3, \text{Class 1}) = 0 - 1.107 = -1.107$$

$$\text{SO}(w_3, \text{Class 2}) = 0 - 1.107 = -1.107$$

**Observation:**

The SO for  $w_3$  and Class 3 is significantly positive, indicating a strong association with the minority class. The negative SO values for other classes reflect the lack of association.

**Constructing Feature Vectors** Consider a document  $d$  containing words  $\{w_3, w_6\}$ .

**Compute SO Values for  $w_6$ :**

First, compute PMI for  $w_6$  and Class 3:

- $p(w_6, \text{Class 3}) = \frac{2}{47} \approx 0.043$
- $p(w_6) = \frac{2}{47} \approx 0.043$
- $p(\text{Class 3}) = 0.1$

Compute PMI:

$$\text{PMI}(w_6, \text{Class 3}) = \log_2 \left( \frac{0.043}{0.043 \times 0.1} \right) = \log_2 \left( \frac{0.043}{0.0043} \right) = \log_2(10) \approx 3.322$$

Compute average PMI for  $w_6$ :

$$\overline{\text{PMI}}(w_6) = \frac{1}{3}(0 + 0 + 3.322) = 1.107$$

Compute SO:

$$\text{SO}(w_6, \text{Class 3}) = 3.322 - 1.107 = 2.215$$

$$\text{SO}(w_6, \text{Class 1}) = 0 - 1.107 = -1.107$$

$$\text{SO}(w_6, \text{Class 2}) = 0 - 1.107 = -1.107$$

**Initialize Feature Vector  $\mathbf{F}_d$ :**

Let  $\mathbf{F}_d = [0, 0, 0]$ , corresponding to Classes 1, 2, and 3.

**Aggregate SO Values:**

For each word  $w$  in  $d$  and each class  $c$ :

- **For  $w_3$ :**

- $\mathbf{F}_d[1] += \text{SO}(w_3, \text{Class 1}) = -1.107$

- $\mathbf{F}_d[2] += \text{SO}(w_3, \text{Class 2}) = -1.107$

- $\mathbf{F}_d[3] += \text{SO}(w_3, \text{Class 3}) = 2.215$

- **For  $w_6$ :**

- $\mathbf{F}_d[1] += \text{SO}(w_6, \text{Class 1}) = -1.107$

- $\mathbf{F}_d[2] += \text{SO}(w_6, \text{Class 2}) = -1.107$

- $\mathbf{F}_d[3] += \text{SO}(w_6, \text{Class 3}) = 2.215$

**Summing the SO Values:**

- $\mathbf{F}_d[1] = -1.107 + (-1.107) = -2.214$

- $\mathbf{F}_d[2] = -1.107 + (-1.107) = -2.214$

- $\mathbf{F}_d[3] = 2.215 + 2.215 = 4.430$

**Normalizing the Feature Vector** Compute the  $L_2$  norm:

$$\|\mathbf{F}_d\|_2 = \sqrt{(-2.214)^2 + (-2.214)^2 + (4.430)^2} = \sqrt{4.903 + 4.903 + 19.613} = \sqrt{29.419} \approx 5.425$$

Normalize each component:

- $F_d[1] = \frac{-2.214}{5.425} \approx -0.408$
- $F_d[2] = \frac{-2.214}{5.425} \approx -0.408$
- $F_d[3] = \frac{4.430}{5.425} \approx 0.817$

### Final Feature Vector

$$\mathbf{F}_d = [-0.408, -0.408, 0.817]$$

**Interpretation** The positive value for Class 3 indicates a strong association with the minority class, while negative values for Classes 1 and 2 reflect a lack of association. This feature vector emphasizes the minority class, enhancing the classifier's ability to detect instances from this class.

**Impact on Classification** When training a classifier with these feature vectors:

- **Enhanced Minority Class Representation:** Documents belonging to the minority class (Class 3) will have higher feature values in the dimension corresponding to Class 3.
- **Better Decision Boundaries:** The classifier can more easily distinguish between classes because the features amplify the signals from the minority class.

This example demonstrates how PMI and SO calculations adjust feature representations to mitigate class imbalance, thereby improving classification performance for minority classes.

### Creation of PMI-SO Feature Vectors

After computing PMI and SO, the next step entails constructing feature vectors for each document using the integrated PMI-SO approach. This is accomplished by aggregating the SO values of the words present in the document, weighted by their

PMI scores. The aggregation can be performed through adding or averaging, depending on the desired representation.

The feature vector for a document  $d$  can be expressed as:

$$\mathbf{F}_d = \sum_{w \in d} \text{SO}(w, c) \cdot \text{PMI}(w, d) \quad (5)$$

Where:

- $\mathbf{F}_d$  is the feature vector for document  $d$ .
- $\text{SO}(w, c)$  is the semantic orientation of word  $w$  with respect to class  $c$ .
- $\text{PMI}(w, d)$  is the PMI between word  $w$  and document  $d$ .

This approach ensures that the feature vectors include both the semantic orientation towards classes and the degree of association between words and documents. The model exclusively utilizes PMI and SO, directly analyzing the statistical relationships inherent in the data without external dependencies.

### 3.3.3 How PMI-SO Features Mitigate Class Imbalance

**Mathematical Explanation** PMI is sensitive to the co-occurrence of words and classes. For minority classes, even if  $p(c)$  is small, a word  $w$  that frequently occurs in  $c$  will result in a high PMI value due to the small denominator  $p(w) \cdot p(c)$ .

SO adjusts PMI by subtracting the average PMI across all classes, further amplifying the significance of words uniquely associated with the minority class. Mathematically:

$$\text{SO}(w, c) = \text{PMI}(w, c) - \frac{1}{|C|} \sum_{c' \in C} \text{PMI}(w, c')$$

**Impact on Feature Space and Classification** By incorporating PMI-SO features, the feature vectors for documents belonging to the minority class accumulate higher

values in the dimensions corresponding to that class. This results in:

- Enhanced representation of minority classes in the feature space.
- Improved separation between classes, aiding classifiers in establishing better decision boundaries.
- Mitigation of bias towards majority classes, commonly observed in imbalanced datasets.

### 3.3.4 Visualization and Empirical Validation

Empirical validations and visualizations, utilizing Principal Component Analysis (PCA) or other dimensionality reduction techniques, demonstrate that PMI-SO features more effectively differentiate minority classes compared to baseline methods. This supports the theoretical claims that these features enhance class discrimination, especially when class distributions are skewed.

#### Dataset for Visualization

A simplified dataset based on the numerical example provided earlier is used. The dataset consists of 10 documents belonging to three classes, with Class 3 being the minority class. Table 3.2 summarizes the documents, their assigned classes, and the words they contain.

Table 3.2: Summary of Documents for Visualization

| Document ID | Class   | Words in Document |
|-------------|---------|-------------------|
| D1          | Class 1 | {w1, w2, w4}      |
| D2          | Class 1 | {w1, w2, w5}      |
| D3          | Class 1 | {w1, w2}          |
| D4          | Class 2 | {w2, w5}          |
| D5          | Class 2 | {w2, w2, w5}      |
| D6          | Class 2 | {w2, w2, w1}      |
| D7          | Class 3 | {w3, w6}          |
| D8          | Class 3 | {w2, w3}          |
| D9          | Class 1 | {w1, w2}          |
| D10         | Class 2 | {w2}              |

## Visualization Methodology

The feature vectors for each document are generated using the PMI-SO approach as previously described. Each document is depicted as a vector in a three-dimensional feature space, where each dimension corresponds to the aggregated SO values for one of the classes.

To visualize the documents in two dimensions, (PCA), a dimensionality reduction technique (B. M. S. Hasan et al. 2021) that projects the data onto the principal components capturing the most variance, is applied.

### Steps of the Visualization Process

1. **Compute SO Values:** Using the SO values calculated in the numerical example, SO values are assigned to each word-class pair. For words not previously calculated, reasonable assumptions are made or their SO values are computed accordingly.
2. **Construct Feature Vectors:** For each document, aggregate the SO values of the words it contains to form a feature vector  $\mathbf{F}_d$ . The vector has one component for each class.
3. **Normalize Feature Vectors:** Each feature vector is normalized using the  $L_2$  norm to ensure comparability between documents.
4. **Dimensionality Reduction:** Apply PCA (or other dimensionality reduction methods) to the normalized feature vectors to reduce the dimensionality from three to two.
5. **Visualization:** The documents are plotted in the two-dimensional space defined by the principal components, with different colors or markers indicating their class labels.

## Explanation of the Visualization

Figure 3.2 illustrates the distribution of the documents in the reduced two-dimensional feature space. Each point represents a document, and the color corresponds to its class label.

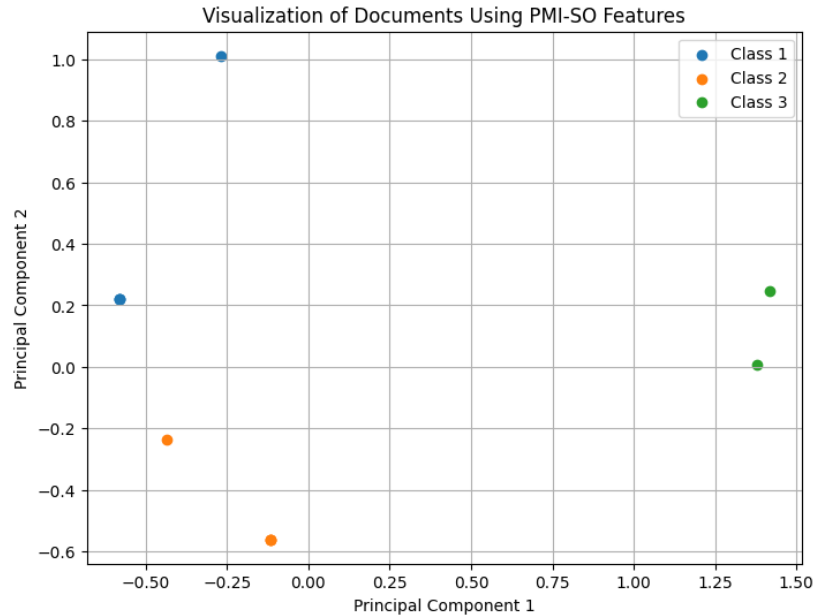


Figure 3.2: Visualization of Documents Using PMI-SO Features and PCA

## Analysis of the Results

From the visualization, the following observations are made:

- **Class Separation:** Documents associated with minority class (Class 3) are distinctly separated from those of Classes 1 and 2. This demonstrates that the PMI-SO features effectively capture the unique characteristics of the minority class.
- **Clustering of Majority Classes:** Documents from Classes 1 and 2 are clustered closely, indicating similarities in their feature representations. Nonetheless, some separation is still evident, suggesting that the PMI-SO features provide discriminative power between these classes.
- **Impact of PMI-SO Features:** The enhancement of features associated with the

minority class improves its representation in the feature space, alleviating the impact of class imbalance and increasing the classifier’s ability to distinguish it from majority classes.

## **Conclusion**

The visualization illustrates that the PMI-SO feature engineering method improves class separability, particularly for the minority class. This empirical validation confirms the theoretical claims about the effectiveness of the methodology in handling imbalanced multi-class text classification.

### **3.3.5 Focus on Document-Level Analysis**

STREAM prioritizes the analysis of documents rather than sentences. This choice is motivated by the necessity to capture the overall semantic context and thematic elements that are often dispersed throughout entire documents. Document-level analysis enables the model to leverage the full scope of available information, which is essential for tasks where the class label applies to the entire document, such as topic classification or sentiment analysis.

By considering the document as a whole, STREAM mitigates data sparsity challenges that may occur when analyzing shorter text segments such as sentences. This approach corresponds with practical applications of text classification, where decisions are generally made at the document level.

### **3.3.6 Handling Multi-Class Classification**

Multi-class classification poses distinct issues compared to binary classification due to the increased complexity in differentiating among multiple classes and the likelihood for overlapping class boundaries. In the context of imbalanced datasets, these issues are intensified as minority classes may be underrepresented, leading to biased classifiers. The PMI-SO approach inherently facilitates multi-class tasks by calculating class-specific SO values for each word. This produces feature vectors that

capture nuanced distinctions among multiple categories simultaneously.

### **STREAM's Approach to Multi-Class Classification**

The STREAM framework addresses these challenges through its feature engineering methodology, which inherently supports multi-class scenarios.

**PMI-SO Features for Multi-Class Representation** For each word  $w$  and class  $c$  in the set of classes  $C$ , the Semantic Orientation (SO) is computed as:

$$\text{SO}(w, c) = \text{PMI}(w, c) - \overline{\text{PMI}}(w)$$

This calculation results in a feature representation where each word has an SO value for every class, capturing its association across all classes simultaneously. The document feature vector  $\mathbf{F}_d$  aggregates these SO values, resulting in a multi-dimensional vector of length  $|C|$ , the number of classes.

**Mathematical Formulation** Given  $K$  classes, the feature vector for document  $d$  is:

$$\mathbf{F}_d = [F_{d,1}, F_{d,2}, \dots, F_{d,K}]$$

Where:

$$F_{d,k} = \sum_{w \in d} \text{SO}(w, c_k)$$

**Discriminative Power and Class Separation** By computing SO values that highlight words uniquely associated with specific classes, STREAM enhances the discriminative power of the feature vectors. This results in better separation between classes in the feature space, allowing classifiers to model complex multi-class decision boundaries more effectively.

## Addressing Class Overlaps and Imbalance

The PMI-SO features adjust for class imbalances by enhancing the representation of minority classes, as discussed in chapter 5. This guarantees that minority classes are not overshadowed by majority classes in the feature space.

**Impact on Classifier Performance** Classifiers trained on PMI-SO feature vectors are better equipped to handle multi-class classification tasks due to:

- **Enhanced Feature Representation:** The multi-class SO values provide rich information about class associations.
- **Improved Decision Boundaries:** The adjusted feature space allows classifiers to establish more accurate boundaries between classes.
- **Sensitivity to Minority Classes:** The amplification of minority class features mitigates the risk of misclassification.

## Empirical Validation

Experiments conducted on multi-class datasets demonstrate the effectiveness of STREAM in handling multi-class classification. Evaluation metrics such as weighted F-Measure and confusion matrices indicate improved performance compared to baseline methods.

### 3.3.7 Classifier Selection

The choice of classifiers is critical in effectively leveraging the semantically enriched feature vectors generated by STREAM. The classifiers selected for this research include:

- **RF:** for its robustness to overfitting and ability to handle high-dimensional data via ensemble learning.
- **SVM:** for its effectiveness in high-dimensional spaces and capability to handle

non-linear decision boundaries using kernel functions.

- **KNN**: for its simplicity and effectiveness, particularly in multi-class classification, with the ability to adapt to imbalanced data through proximity-based weighting.
- **NB**: for its simplicity and efficiency, particularly suitable for text classification due to its probabilistic nature.
- **DT**: for its interpretability and ability to model complex decision boundaries in multi-class scenarios.

Parameters for each classifier are carefully adjusted to optimize performance. For instance, in the case of SVM, the regularization parameter  $C$  and kernel parameters are adjusted through cross-validation to achieve the best trade-off between margin maximization and classification accuracy. For RF, the number of trees and depth are set to balance between computational efficiency and model performance. Further details of these classifiers are presented in Section 2.3.3 (Classification Algorithms).

### 3.3.8 Feature Normalization and Scalability Considerations

Normalization procedures are employed to guarantee that the feature vectors facilitate good classification. Standardization methods such as z-score normalization are used to adjust features so that they have a mean of zero and a standard deviation of one (Jain et al. 2005). This step is crucial when utilizing classifiers sensitive to feature scaling, such as SVM.

By normalizing the PMI-SO feature vectors, the impact of scale differences between features is mitigated, allowing the classifiers to perform optimally.

Normalization ensures that all features contribute equally to classification. Regarding scalability, as the size of the dataset and vocabulary grows, computing PMI and SO for all words and classes can increase in computational cost. Efficient data

structures, parallelization, and vocabulary reduction strategies can be considered to maintain tractability. While neural embeddings often require powerful GPUs and large-scale pretraining, STREAM’s reliance on co-occurrence statistics is relatively straightforward and can occasionally be scaled in a more cost-efficient manner.

### 3.4 STREAM Pseudocode

The STREAM methodology described in the preceding sections consists of several interconnected stages, beginning with data acquisition and preprocessing, followed by feature construction using Pointwise Mutual Information and Semantic Orientation, and concluding with classifier training and evaluation. To improve clarity, the overall workflow is summarised below in pseudocode form. The pseudocode is intended to provide a high-level procedural view of the proposed methodology rather than a software-specific implementation. Its purpose is to show the logical sequence of the major stages involved in STREAM, while the detailed mathematical definitions, design choices, and evaluation procedures remain as described in the surrounding sections.

#### 1. Data Acquisition:

Collect raw textual data  $D$  along with class labels  $C$ .

#### 2. Preprocessing:

document  $d \in D$  a. **Normalization:**

- i. Convert text to lowercase.
- ii. Remove punctuation, numbers, special characters.

b. **Tokenization:** Split text into tokens/words.

#### c. Optional Steps:

- i. Remove stop words.
- ii. Apply lemmatization or stemming.

#### 3. Feature Engineering using PMI and SO:

a. Build vocabulary  $V$  from all tokens in  $D$ .

b. Compute term frequencies and class distributions:

word  $w \in V$  Compute frequency  $f(w)$ . class  $c \in C$  Compute frequency  $f(c)$ . c. **Compute**

**PMI:**

word  $w \in V$  and class  $c \in C$  i. Compute joint frequency  $f(w, c)$ .

ii. Compute joint probability  $p(w, c) = \frac{f(w, c)}{N}$ , where  $N$  is total word occurrences.

iii. Compute marginal probabilities:

$$p(w) = \frac{f(w)}{N}$$

$$p(c) = \frac{f(c)}{|D|}, |D| \text{ is total documents.}$$

iv. Compute  $\text{PMI}(w, c) = \log_2 \left( \frac{p(w, c)}{p(w) \cdot p(c)} \right)$ .

d. **Compute Semantic Orientation (SO):**

word  $w \in V$  and class  $c \in C$  i. Compute average PMI over all classes:

$$\overline{\text{PMI}}(w) = \frac{1}{|C|} \sum_{c' \in C} \text{PMI}(w, c').$$

ii. Compute  $\text{SO}(w, c) = \text{PMI}(w, c) - \overline{\text{PMI}}(w)$ .

e. **Create PMI-SO Feature Vectors:**

document  $d \in D$  Initialize feature vector  $\mathbf{F}_d$  of length  $|C|$ .

word  $w \in d$  class  $c \in C$   $\mathbf{F}_d[c] += \text{SO}(w, c)$ .

Normalize  $\mathbf{F}_d$  (e.g., using  $L_2$  norm).

#### 4. Data Splitting:

Split  $D$  into training set  $D_{\text{train}}$  and test set  $D_{\text{test}}$ .

#### 5. Classifier Training:

classifier  $M$  in {RF, SVM, KNN, NB, DT} a. Train  $M$  using  $D_{\text{train}}$  and feature vectors  $\mathbf{F}_d$ .

b. Tune hyperparameters via cross-validation.

#### 6. Model Evaluation:

trained classifier  $M$  a. Predict class labels for  $D_{\text{test}}$  using  $M$ .

b. Compute evaluation metrics:

i. Cohen's Kappa  $\kappa$ .

ii. Weighted F-Measure.

iii. AUC (for binary classification).

- iv. Additional metrics: Accuracy, Precision, Recall.
- c. Analyze performance across all classes.

#### **7. Output:**

Select best-performing model(s) based on evaluation metrics.  
Provide insights and interpretations of the results.

## **3.5 Technical Discussion on Methodological Choices**

The methodological choices made in STREAM are grounded in both theoretical considerations and practical effectiveness.

### **3.5.1 Justification for Using PMI and SO**

PMI is employed due to its ability to capture the strength of association between words and between words and documents, which is more informative than simple co-occurrence counts. It effectively highlights word pairs and word-document associations that occur together more frequently than expected by chance, uncovering meaningful semantic relationships.

SO extends this by focusing on the association of words with specific classes. By calculating SO, words that are particularly indicative of certain classes can be identified, which is essential for classification tasks. This is especially important in multi-class and imbalanced scenarios, where highlighting features associated with minority classes can improve model performance.

### **3.5.2 Creation of Feature Vectors Using PMI-SO**

The decision to create feature vectors using integrated PMI-SO approach is motivated by the intention to directly use statistical relationships within the data. By combining PMI and SO, the feature vectors reflect both the strength of word associations and their class-specific orientations. This results in a rich representation that

captures the essential semantic characteristics required for efficient classification.

This approach avoids reliance on external embedding models or frequency-based features like TF-IDF, which may not adequately capture class-specific semantic nuances, especially in imbalanced datasets.

### 3.5.3 Emphasis on Feature Engineering Over External Embeddings

STREAM prioritizes feature engineering based on PMI and SO over the use of external embeddings such as Word2Vec. This choice is motivated by several factors:

- **Task-Specific Representations:** PMI-SO features are derived directly from the dataset at hand, ensuring that the representations are tailored to the specific classification task.
- **Interpretability:** The statistical basis of PMI and SO allows for more interpretable models, facilitating understanding of how certain features contribute to classification decisions.
- **Computational Efficiency:** By avoiding complex neural embedding models, STREAM reduces computational overhead, making it practical for large-scale or resource-constrained environments.

### 3.5.4 Rationale for Classifier Choices and Parameters

The choice of classifiers is determined by their appropriateness for multi-class and binary text classification tasks and their compatibility with the PMI-SO feature space generated by the STREAM methodology. Each chosen classifier represents a distinct learning paradigm—probabilistic, distance-based, kernel-based, tree-based, and ensemble-based—offering a broad perspective on how the derived features affect classification performance. The parameters are chosen to create robust, representative baselines with minimal tuning, ensuring that observed improvements can be more confidently attributed to the STREAM feature engineering rather than intricate model

optimization. A review of the literature on these algorithms is presented in Section 2.3.3 (Classification Algorithms)

**NB:** NB classifier operates under conditional independence assumptions. Its default parameter settings are employed, representing a straightforward and computationally efficient baseline. NB often excels in textual domains, and the probabilistic nature of NB aligns naturally with the PMI-SO features, which encode semantic orientation towards different classes.

**DT:** A decision tree model (using the J48 variant of the C4.5 algorithm) is selected for interpretability and direct modeling of non-linear relationships. Several parameters are critical for ensuring robust model performance. The confidence factor is set to 0.25, facilitating more assertive pruning and mitigating overfitting by trimming overly complex branches. The minimum number of instances per leaf is set to 2, enabling the tree to generate both specialized and generalizable rules. Subtree raising is activated to enhance the pruning process further, and the Minimum Description Length correction is applied to penalize unnecessary complexity. Collectively, these parameters maintain a balance between model expressiveness and generalization, ensuring that the features identified by STREAM or feature selection methods yield significant, stable partitions in the feature space.

**KNN:** KNN is a non-parametric, instance-based algorithm that offers a valuable comparison to more complex classifiers. The dependence on closeness within the feature space enables a direct evaluation of the quality of the PMI-SO features: if these features accurately represent semantic distinctions, documents from the same class should cluster, leading to improved classification performance. To ensure good generalization, the algorithm selects the optimal number of neighbors  $K$  automatically via cross-validation on the training set. This process determines the value of  $K$  that yields the best validation performance, thereby avoiding arbitrary parameter choices and highlighting the impact of the feature representation on neighborhood quality.

**SVM:** SVMs are selected for their robustness in high-dimensional environments and their ability to model complex, non-linear boundaries—attributes valuable in handling nuanced textual semantics captured by PMI-SO. Various kernel functions, including the Radial Basis Function (RBF), Polynomial, and Linear kernels, were evaluated on the test sets to identify the most effective kernel. Despite being fundamentally binary classifiers, common multi-class techniques, such as one-against-one or one-against-all approaches, permit SVMs to manage multi-class classification. The selected kernels and basic default parameters provide a fair assessment of how well SVM leverages the semantically informed features, rather than focusing on extensive parameter tuning.

**RF:** RF is an ensemble technique that integrates multiple decision trees, offering robustness against overfitting and the ability to manage large, high-dimensional feature sets. In this configuration, the RF model generates 100 trees, ensuring adequate model capacity. Each tree is trained on a bootstrap sample of the data (`bagSizePercent=100%`), maintaining data diversity and reducing variance. Trees grow without a maximum depth limit (`maxDepth=0`), allowing them to capture complex patterns, though the ensemble framework inherently reduces the risk of severe overfitting. The default feature-split heuristic (often the square root of the number of features) improves tree diversity, while functionalities like out-of-bag error estimation and feature importance computation are disabled to maintain focus on classification performance. This setup balances computational demands with the objective of evaluating how well the derived features support accurate predictions.

### **Summary of Rationale**

The selected classifiers—NB, DT, KNN, SVM, and RF—embody a varied array of learning methods. The parameters of each classifier are chosen to be representative and stable, minimizing the necessity for costly hyperparameter searches that can obscure the primary contribution of the STREAM technique. This approach ensures that the improvements observed across baseline and STREAM configurations can

be confidently attributed to the quality and semantic richness of the engineered features, rather than the complexities of model parameter optimization.

### **3.5.5 Positioning Against State-of-the-Art Approaches**

Although powerful, pretrained embeddings (e.g., Word2Vec, GloVe, BERT) often rely on extensive corpora for training (Kim et al. 2021) and may not inherently address class imbalance. These embeddings encapsulate profound contextual semantics but lack inherent class-sensitivity without additional modification. Conversely, STREAM is designed from the ground up to highlight class distinctions, especially for minority classes.

Furthermore, neural approaches may incur significant processing costs and necessitate thorough optimization. The PMI-SO method is more straightforward to implement, comparatively easier to scale to new domains, and highly interpretable. Researchers and practitioners can combine STREAM with neural embeddings, potentially using PMI-SO features to direct attention towards minority-class indicators or to serve as a complementary signal in hybrid models.

## **3.6 Performance Evaluation**

The evaluation strategy adopted in this thesis is designed to provide unbiased, reproducible, and comparable estimates of generalisation performance across multiple datasets, representations, and learning configurations. Given the central role of model comparison in the empirical validation of the proposed approach, particular care is taken to ensure that performance estimates are not inflated by inappropriate reuse of data during model selection, nor distorted by class imbalance or stochastic variation in experimental conditions. To this end, the evaluation framework combines cross-validated estimation, stratified data partitioning, class-sensitive performance metrics, and controlled execution protocols that explicitly account for computational constraints.

At a high level, each experiment proceeds by repeatedly partitioning the available labelled data into disjoint subsets for training, validation, and testing, fitting models exclusively on training data, selecting hyperparameters using validation data only, and finally assessing predictive performance on unseen test data. This separation between model construction and model assessment is a foundational requirement for reliable evaluation in supervised learning, particularly when multiple configurations are compared under a shared experimental setup.

### 3.6.1 Experimental Execution Protocol

All experiments are executed through a unified experimental pipeline that enforces consistent data handling, transformation, and evaluation across datasets. For a given dataset, representation, and classifier configuration, the pipeline automatically performs data splitting, feature transformation, hyperparameter optimisation, and metric computation according to predefined evaluation settings. Randomisation is controlled through fixed seeds to ensure that results are reproducible and that comparisons between models are not confounded by differences in data partitioning.

Although nested cross-validation is widely regarded as a rigorous strategy for model selection and performance estimation, particularly when hyperparameter tuning is involved, it is also computationally demanding in large-scale text classification settings because the optimisation procedure must be repeated across multiple training folds, parameter combinations, and classifier configurations (Cawley et al. 2010). In this thesis, a controlled fallback configuration, termed *emergency mode*, is therefore introduced for cases in which the default optimisation procedure become computationally prohibitive due to runtime and memory constraints associated with high-dimensional feature spaces and repeated fold-based evaluation. This configuration is adopted for practical computational feasibility rather than as a change to the methodological principles of the study.

Emergency mode does not alter the underlying datasets, class distributions, feature construction process, or evaluation metrics. Instead, it reduces the computational depth of model selection in a controlled and transparent manner, while preserving the core safeguards of the experimental design, including fold-safe preprocessing, class stratification, and strict separation between training, validation, and testing data. This is important because the main objective of the fallback configuration is to maintain a leakage-free and comparable evaluation framework even when full optimisation is not computationally feasible. The use of such a constrained search setting is consistent with the broader model-selection literature, which recognises that hyperparameter optimisation often operates under finite computational budgets and that reduced search can be a legitimate compromise when applied systematically and without violating evaluation integrity (Bergstra et al. 2012; Cawley et al. 2010).

To preserve fairness and comparability across datasets, a strict standardisation rule is enforced: if emergency mode is required for any dataset under a given experimental configuration, the same emergency configuration is applied uniformly across all datasets in that experimental group. This prevents situations in which some datasets benefit from more extensive optimisation than others and ensures that observed performance differences reflect properties of the models and data rather than unequal computational treatment. All results reported in this thesis are therefore derived either entirely from default-mode experiments or entirely from emergency-mode experiments, depending on feasibility, and the mode used is explicitly documented.

### **3.6.2 Cross-Validation and Model Selection Strategy**

The primary mechanism for performance estimation in the default configuration is nested, stratified cross-validation. Nested evaluation is employed to address the well-documented bias that arises when the same cross-validation procedure is used both to select hyperparameters and to estimate generalisation performance. In this

framework, the data are first partitioned into a set of outer folds, which are used exclusively for performance assessment. Within each outer training partition, a further stratified cross-validation procedure is applied to select hyperparameters.

Stratification is used consistently at all stages of data partitioning to preserve the original class proportions within each fold. This is particularly important given the class imbalance present in several of the datasets considered. The number of folds in both the outer and inner procedures is fixed by default but is dynamically constrained by the minimum per-class sample count to avoid degenerate splits in which one or more classes are absent from a fold. This safeguard ensures that all computed metrics remain well-defined and that per-class performance is meaningfully estimated.

Within the inner cross-validation procedure, hyperparameter optimisation is conducted using either exhaustive grid search or an adaptive resource allocation strategy based on successive halving, depending on the experimental setting. When successive halving is employed, computational resources are allocated preferentially to promising configurations while eliminating weaker candidates early, following established principles of efficient hyperparameter optimisation. Regardless of the search strategy used, all transformations, including feature extraction and feature selection, are fitted exclusively on the training portion of each fold and applied to validation or test data only after fitting, thereby preventing any form of data leakage.

### **3.6.3 Holdout Evaluation**

In addition to cross-validated estimation, a stratified holdout evaluation is performed to provide a single, deployment-oriented performance estimate. In this setting, each dataset is split once into training and test partitions using an 80/20 stratified split with a fixed random seed. Hyperparameter selection is carried out exclusively on the training partition, after which the selected model is refit on the full

training data and evaluated on the held-out test set. While this evaluation does not replace cross-validation as the primary source of performance estimates, it serves as a complementary check on the stability and plausibility of the cross-validated results.

To characterise uncertainty in holdout performance, nonparametric bootstrap resampling is applied to the test predictions, and confidence intervals are computed for the weighted F-measure. Bootstrap-based intervals are well suited to this context because they make minimal distributional assumptions and provide an empirical estimate of variability induced by sampling.

### 3.6.4 Evaluation Metrics

Performance is evaluated using a set of metrics chosen to reflect both overall predictive accuracy and sensitivity to class imbalance. The primary metric used for model selection and reporting is the weighted F-measure ( $F_1^{(w)}$ ), defined as the class-frequency-weighted average of per-class  $F_1$  scores. The  $F_1$  measure, as the harmonic mean of precision and recall, penalises models that achieve imbalanced trade-offs between these quantities. Weighted averaging is used to ensure that performance on minority classes contributes proportionally to the overall score, while still reflecting the empirical class distribution.

In addition to  $F_1^{(w)}$ , weighted precision, weighted recall, and overall accuracy are reported to provide complementary perspectives on model behaviour. Accuracy is included primarily for comparability with prior work but is not treated as a sufficient indicator of performance in imbalanced multi-class settings. To further account for chance agreement induced by skewed class priors, Cohen’s  $\kappa$  is also computed for each evaluation run. Where appropriate, per-class metrics are examined to identify systematic error patterns, although detailed per-class reporting is restricted to datasets with a manageable number of classes.

### **3.6.5 Aggregation, Variability, and Comparative Analysis**

For cross-validated experiments, performance metrics are computed separately for each outer fold and aggregated by arithmetic mean to produce a single estimate per configuration. Fold-wise results are retained to capture variability induced by data partitioning and to support paired statistical comparisons. Although cross-validation reduces variance relative to single-split evaluation, fold-wise estimates are not statistically independent, and variance estimates are interpreted accordingly.

When formal comparisons between models are required, paired nonparametric tests are applied to fold-wise scores using the Wilcoxon signed-rank test. This test is appropriate in the present context because it operates on matched observations and does not assume normality of score differences. All comparisons are conducted on identical data partitions, ensuring that observed differences reflect model behaviour rather than differences in evaluation conditions, in line with established recommendations for classifier comparison.

### **3.6.6 Evaluation Integrity and Limitations**

The evaluation framework enforces strict separation between training, validation, and testing at all stages of experimentation. No information from evaluation partitions is permitted to influence feature extraction, feature selection, or hyperparameter tuning. While nested cross-validation substantially reduces selection bias, it does not eliminate all sources of dependency between estimates, and reported results are interpreted with this limitation in mind. Similarly, the use of emergency mode represents a pragmatic compromise between computational feasibility and methodological rigour; however, by standardising its application across datasets and preserving all evaluation principles, the integrity and comparability of results are maintained.

## 3.7 Conclusion

This chapter has presented the methodological foundations of the proposed STREAM (Semantically Oriented Vector-Based Representation) framework and the experimental design adopted to evaluate it rigorously. The objective of the methodology has been twofold: first, to formalise the construction of class-aware semantic representations that address multi-class imbalance in text classification; and second, to establish a reproducible and statistically principled evaluation protocol capable of assessing STREAM against strong baseline approaches under controlled conditions.

From a modelling perspective, STREAM was formulated as a representation-learning strategy that integrates semantically oriented vector construction with predictive feature augmentation, ensuring that minority-class signals are preserved during feature generation rather than suppressed by global frequency effects. The methodology explicitly separates representation construction from classifier choice, enabling STREAM to be evaluated across multiple learning algorithms without conflating representational and algorithmic contributions. This separation is essential for isolating the source of performance gains and for ensuring fair comparison with Bag-of-Words and Word2Vec baselines.

From an experimental perspective, the evaluation design was constructed to minimise bias and maximise comparability. Nested, stratified cross-validation was adopted as the primary estimation strategy in order to prevent optimistic bias during hyperparameter tuning. A complementary stratified holdout protocol was included to provide deployment-oriented performance estimates and bootstrap-based uncertainty quantification. All data partitioning procedures preserve class distributions, and safeguards are implemented to avoid degenerate folds in the presence of minority classes. Furthermore, paired nonparametric statistical testing is incorporated to support robust comparative claims between models.

Recognising the computational demands associated with extensive hyperparam-

eter optimisation across multiple datasets and configurations, the methodology also formalises an emergency execution mode. This mode provides a controlled reduction in optimisation complexity when resource constraints prevent full execution of the default configuration. Crucially, its application is standardised across datasets to preserve fairness and comparability, thereby maintaining methodological integrity even under constrained computational conditions.

The proposed STREAM methodology is evaluated across heterogeneous benchmark datasets, including Cyberbullying detection data, the New York Times corpus, Reuters newswire, Yelp reviews, and the 20-Newsgroups collection. These datasets span informal and formal text, short and long documents, and varying degrees of class imbalance, enabling a comprehensive assessment of STREAM’s generalisation behaviour across domains and difficulty regimes.

Collectively, the modelling framework and evaluation protocol established in this chapter provide a rigorous foundation for empirical analysis. By enforcing strict separation between training, validation, and testing, controlling for class imbalance, standardising execution procedures, and supporting statistically grounded comparisons, the methodology ensures that the results reported in subsequent chapters reflect genuine differences in representational capacity rather than artefacts of experimental design. The following chapters present and analyse these empirical findings in detail, examining STREAM’s behaviour under both standard and stress conditions across all considered datasets.



## 4 | Cyberbullying Severity Detection

This chapter provides the first empirical grounding of the STREAM methodology introduced in the preceding chapter. The Cyberbullying dataset is used here as an initial validation environment because it exhibits two properties that are central to the thesis problem setting: a strongly imbalanced multi-class label distribution and fine-grained semantic boundaries between adjacent classes. The task is formulated exclusively as four-class severity prediction, with labels *none*, *low*, *medium*, and *high*.

The chapter has two complementary aims. The first is to establish how far conventional text representations, namely Bag-of-Words and Word2Vec, can progress under the thesis evaluation protocol when combined with standard learners. The second is to evaluate whether STREAM’s semantic orientation mechanism provides measurable benefit on a dataset where harmful intent is often expressed indirectly and severity distinctions are subtle. This is a deliberately demanding starting point: the difference between *low* and *medium* severity is frequently a matter of implication and tone rather than explicit lexical markers, while the *high* class is comparatively sparse and therefore sensitive to both data scarcity and leakage-prone evaluation practices.

All experiments reported in this chapter follow the fold-safe, pipeline-based evaluation framework adopted throughout the thesis. Because STREAM feature construction and model selection are computationally demanding, the Cyberbullying experiments are executed under the emergency-mode configuration described in the Performance Evaluation chapter. Emergency mode preserves the core method-

ological safeguards, including nested model selection and strict separation of training and evaluation data within each fold, while reducing fold counts to ensure feasible execution. Results are reported using accuracy and weighted  $F_1$  as primary summary measures, alongside agreement measures where appropriate, since accuracy alone can be dominated by the majority *none* and *low* classes in this setting.

A further motivation for this chapter is to examine stability under harsher conditions than the base evaluation. In addition to the main evaluation results, mandatory stress tests are conducted by retaining only a fraction of the available data (keep=0.6 and keep=0.4). These settings amplify minority-class scarcity and allow direct examination of whether relative ranking and class-wise behaviour persist when supervision is reduced. Confusion-matrix analysis is used throughout to interpret error structure, with particular attention to whether misclassifications primarily reflect adjacent-severity ambiguity or systematic absorption of minority classes into majority labels.

Finally, predictive features motivated by prior cyberbullying literature are treated as part of the primary representation rather than auxiliary inputs. These cues are integrated into the main feature space and subjected to the same fold-safe processing and model selection as lexical and semantic features. This ensures that comparisons between STREAM and non-STREAM baselines reflect consistent pipelines and that all reported outcomes correspond to what was actually executed under the thesis evaluation framework.

## **4.1 Data Input**

### **4.1.1 Data Accessibility, Collection, and Annotation**

The successful application of the STREAM methodology begins with securing high-quality, well-annotated datasets. For the cyberbullying detection task, Twitter was chosen as the data source because it is an OSN platform that provides researchers

with access to public tweets. Through Twitter’s application programme interface (API), each tweet may be retrieved and provides an extensive amount of information (Morstatter et al. 2013), including user ID, username, brief bio, account creation date, and tweet content (which contains details about feelings, attitudes, behaviours, and other personally relevant information) (H. Kwak et al. 2010), tweet creation time, tweet’s unique ID, tweet language, the total number of tweets of a user, number of favourites, number of followers, number of mentions, number of following, number of retweets, latitude, and longitude (geo-location), and information about the application that sent a tweet (e.g. iOS, Android). Utilising this information, special features have been created from Twitter data to categorise different applications (Bollen et al. 2011; Eichstaedt et al. 2015; Imran et al. 2016; Java et al. 2007; Preoţiuc-Pietro et al. 2015; Sakaki et al. 2010; Rui Zhao et al. 2016). As discussed below, some of this information is utilized in STREAM to develop a machine learning model to detect cyberbullying severity levels: low, medium, high, or none.

Geo-tagged filtering was applied in the data collection process (Al-garadi et al. 2016; Morstatter et al. 2013) to minimize potential bias in sampling size in the given dataset (Cheng et al. 2014; González-Bailón et al. 2014; Morstatter et al. 2013), and the USA<sup>1</sup> was chosen as the country, based on the highest number of active users, more precisely the state of Texas. The collected dataset contains approximately 2.2 million tweets based on a lexicon provided by (Rezvan et al. 2018) related to the five topics: sexuality, racism, physical- appearance, intelligence, and politics. Tweets were collected between November 2019 and December 2019 according to Twitter’s privacy policies<sup>2</sup>.

#### **4.1.2 Manual Data Annotation**

The main goal of an effective cyberbullying detection system in an OSN is to prevent or at the very least, decrease incidences of harassment and bullying (Einarsen et

---

<sup>1</sup><https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/>

<sup>2</sup><https://developer.twitter.com/en/developer-terms/agreement-and-policy.html>

al. 2002). These detection approaches can serve as tool to aid with and facilitate the supervision of such online settings. Cyberbullying detection may be used to monitor and trace the bully as well as to assist and counsel the victim (Maral Dadvar et al. 2012).

Around 14,495 tweets were randomly selected from the collected dataset (2.2 million tweets) for a manual labelling process, similar to the approach applied in (Al-garadi et al., 2016; Balakrishnan et al., 2020). The participants (labellers) of the study labelled these tweets according to the following labels: low, medium, high, or non-cyberbullying. Tweets were labelled by three postgraduate students with research background in multi-disciplinary fields. To begin, participants (labellers) were trained via conference calls by the author of this thesis in twitter content and indications of the presence of cyberbullying behaviour in tweets. For example, the participants were trained on abbreviations, slang words, and acronyms commonly used in social networks. The participants were then instructed to label whether a tweet contained cyberbullying behaviour or not. If a tweet contained cyberbullying behaviour, then participants were asked to label how severe that cyberbullying was based on their opinion (low, medium, high).

In contrast to the previous studies (Potha et al. 2014; Reynolds et al. 2011), for this thesis research, it was proposed to categorise severity in three levels, by categorising the topics already declared as sensitive and severe, namely: sexuality, racism, physical-appearance, intelligence, and politics. By doing so it was hoped to find how a machine learning multi-class algorithm for detecting cyberbullying might perform. Inspired by (Duggan 2014), in order to study severity levels in a on-line social network, the above-mentioned forms of cyberbullying were categorised into three levels: low, medium, high, and non-cyberbullied tweet. Moreover, a voting scheme between the participant (labellers) was adopted (Al-garadi et al. 2016) to determine whether a tweet involved cyberbullying and, if so, its severity level. Specifically, a tweet was considered cyberbullying if at least two of the assigned labellers identified it as such. The same voting scheme was used to establish con-

sensus on the severity level of the cyberbullying. If there was no agreement among the labellers, the tweet was discarded from the dataset.

Initially, the dataset contained 14,495 tweets. However, during the annotation process, tweets without a clear consensus among the labellers were excluded. This resulted in a reduction of the dataset to 11,904 tweets. The final dataset was classified as follows: low severity (404 tweets, 3%), medium severity (590 tweets, 5%), high severity (443 tweets, 4%), and non-cyberbullying (10,467 tweets, 88%). This process ensured that only tweets with a high degree of agreement among labellers were included in the final dataset, thereby improving the reliability and accuracy of the annotations.

After the manual annotation (Figure 4.1), the final dataset contained 11,904 tweets classified in the following way: low (404, 3%), medium (590, 5%), high (443, 4%), and non-cyberbullying (10467, 88%).

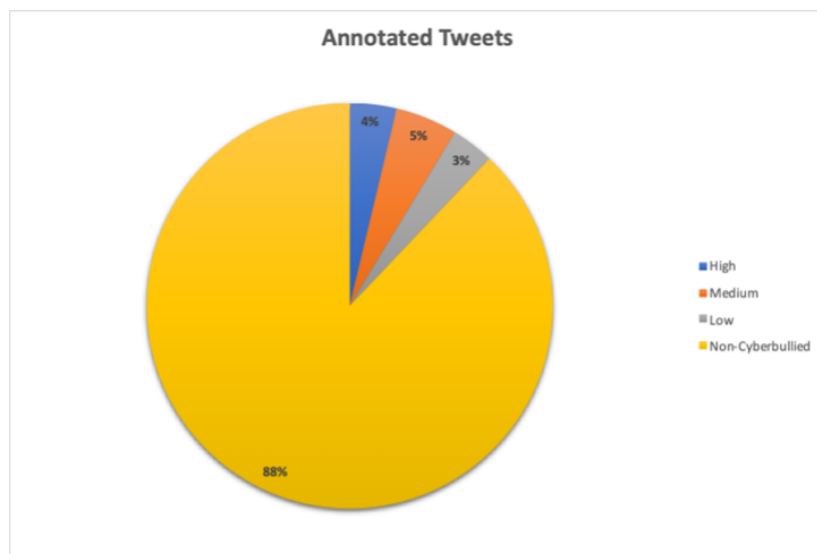


Figure 4.1: Distribution of Tweets by Cyberbullying Severity

### 4.1.3 Preprocessing

Tweets were converted to lower case to avoid any sparsity issue, reduced repeated letters, standardized URLs, and @usermention to remove noise in the tweets. In the BoW approach, tokenization was applied with a Twitter-specific tokenizer based on

the CMU TweetNLP library (Gimpel et al. 2011) and only words with a minimum frequency of 10 were kept.

#### **4.1.4 Feature Extraction**

All tweets were represented in a BoW, Word2Vec, and in manually engineered vector-based to further refine the dataset, a part-of-speech (POS) tagging step was applied using the CMU TweetNLP tagger (Gimpel et al. 2011). POS information can clarify word senses and improve the interpretability of PMI and SO features by offering contextual clues about how each token functions syntactically.

In STREAM, the principle transformation entails deriving semantically enriched features guided by PMI and SO. In contrast to solely frequency-based representations (e.g., Bag-of-Words) or generic embeddings (e.g., Word2Vec), STREAM’s feature engineering prioritizes class-specific associations. By identifying words with strong, class-oriented semantic leanings, STREAM improves the classifier’s ability to identify subtle distinctions among the four severity levels of cyberbullying.

With this preprocessed and annotated dataset in hand, the further steps of STREAM—calculating PMI, determining SO, and constructing final feature vectors—can be performed. The careful preparation guarantees that the methodological pipeline is optimally aligned to achieve the goal of improved multi-class, imbalanced text classification performance.

## **4.2 Feature Generation**

Initially, document-level categorization was employed to assess the SO of each word in the corpus. Phrases were extracted for classification using the POS tags. After phrases were taken out of the dataset, their semantic orientation towards cyberbullying or against cyberbullying was established. The score was calculated by subtracting the PMI of the target word with a cyberbullying class from the PMI of the target word with a non-cyberbullying class. This method was well suited for the

domain-specific lexicon generation with a PMI score (P. Turney 2002). Based on this, domain-specific lexicon with PMI-SO was created for each word and phrase for each class using Turney’s technique (Equation 1-3).

## **4.3 Feature Engineering**

As outlined in Chapter 3 (Section 3.7), feature engineering is the process of generating or deriving features from raw data or corpus, or it is the creation of supplementary features inferring from existing features (Pattnaik et al. 2018) and it is not the number of features, but the quality of the features that are fed into a machine learning algorithm that directly influences the outcome of the model prediction. By its nature, especially in text classification, imbalanced data is high-dimensional, and feature selection methods are critical to achieving optimal performance. Using STREAM, tweets are represented in low-dimensional space with just a single value of SO for each tweet.

Moreover, in the context of the cyberbullying detection model, the most common approaches to improve cyberbullying detection is to perform feature engineering, and the most common features that improve the quality of cyberbullying detection classifier performance are: textual, social, user, sentiment, word embeddings features (Rosa et al. 2019). The selected features include network-based, activity, content, user, personality, and our manually created master feature PMI-SO. To the author’s knowledge, no prior research has utilized the semantic orientation of individual words in tweets as an input feature. The selected features are discussed in more detail in the following sub-sections.

### **4.3.1 Network-Based Features**

Considering the immense popularity of social media and its role as a key source of information for research on the detection of cyberbullying, it should come as no surprise that network data, such as the quantity of friends and follows, uploads, likes,

and other attributes, is being employed more and more frequently as a feature in detection systems (Salawu et al. 2018). Moreover, these features have been proven to be efficient in identifying cyberbullying behaviours in OSN by (Al-garadi et al. 2016; Chatzakou, Vakali, et al. 2017; Navarro et al. 2012). Moreover, network-based properties are the most crucial characteristics to spot violent or cyberbullying behaviours, claim (Chatzakou, Kourtellis, et al. 2017).

### **4.3.2 Activity Features**

Features in this category include the number of tweets a user has posted, the age of a user's account (i.e., number of days since it is created), favoured tweets, URLs, hashtags, and mentioned users (i.e., @username) (Chatzakou, Kourtellis, et al. 2017). (Galán-García et al. 2014) have proposed features based on tweeted text, posting time, language, and location of a user to improve the identification of the user. According to the authors, these features help in the identification and classification of users since they exhibit specific behaviours that may be inferred from an analysis of these variables. For the cyberbullying case study, tweet time is grouped into four groups: midnight, morning, afternoon, and night.

### **4.3.3 User Features**

Limited cyberbullying research has used either social or an individual user's features. There are two main reasons for this. Firstly, there is little information available to researchers about the gender, age, social class, race, geographical location, and so on, of a user. Secondly, many websites do not provide open access to the users' profile data in order to secure user information (e.g. age and gender) from public exploitation. For example, to prevent data of their users from being misused for entertainment purposes (Rosa et al. 2019). However, improvements in NLP technology have made it possible for researchers to automatically compare the many linguistic features in various text styles and determine a user's age and gender (Peersman et al. 2011).

(Hosseini et al. 2016) also predicted age and gender in social networks using text analysis and NLP on a user's messages. Many studies have focused on the parts of speech used in text content, demonstrating that females often use more single pronouns and verbs while males frequently utilise articles and nouns. (M. Dadvar et al. 2012) found that adding gender-based information as a feature increased the overall accuracy of the cyberbullying detection algorithm. In a survey of researchers studying cyberbullying, it was suggested to take gender as a feature into account while creating machine learning models (Van Royen et al. 2015). Women are less prone than men to engage in cyberbullying practises, according to a few research that have looked at the relationship between gender and cyberbullying (O'Connor 2009; Zsila et al. 2019), however, some studies have shown the opposite (Dilmaç 2009; Kowalski et al. 2012; Sourander et al. 2010). Furthermore, it has been shown that adults use more plural pronouns and future tense verbs than children when trying to predict age using NLP (Chung et al. 2007; Schwartz et al. 2013). (Slonje et al. 2008) found that although cyberbullying victims do not all fall into a certain age bracket, teens exhibit the most cyberbullying behaviours, and as people become older, their likelihood of engaging in such behaviour declines.

To generate predicted gender features, an open vocabulary provided by (Schwartz et al. 2013) for 100 most common words used by male/female has been used for the cyberbullying case study. The dataset was also labelled to predict whether a user was a male or a female based on the name reported in their username or tweet (liu)(W. Liu et al. 2013). Similarly, the predicted age feature was generated by using 100 most common words used in social media that were positively or negatively correlated with each age level (Schwartz et al. 2013), specifically, age group 1: 13-18 years; age group 2: 19-22 years; age group 3: 23-29 years; and age group 4: 30 and above.

#### **4.3.4 Content-Based Features**

Because cyberbullying texts are typically rude and disrespectful in nature, it is not surprising that profanity is the most often cited content-based element in the text as a predictor of cyberbullying (Salawu et al. 2018). According to (Q. Huang et al. 2014), by combining social network data such network, activity, and user features with content information, the identification of cyberbullying in OSN may be considerably enhanced.

#### **4.3.5 Personality Features**

Individuals share their thoughts, emotions, and perspectives on OSNs, providing a substantial and enlightening collection of personal information, which serves as a representation of various aspects of their real-world existence, including their individuality. As a result, the proliferation of these online platforms during the last decade presents the text as a prominent candidate for assessing an individual's personality (Carducci et al. 2018). Personality is a distinguishing set of behavioural traits that differentiates individuals from one another. The study conducted by (Mahmud et al. 2013) examined the utilisation of social media content as a means of evaluating an individual's personality traits.

Research has indicated that an individual's personality can be automatically inferred from their communication style (Balakrishnan et al. 2019). Moreover, empirical research has revealed significant associations between the Big Five personality traits and the occurrence of cyberbullying (Resett et al. 2017). The Big Five personality traits encompass five distinct dimensions: extraversion, neuroticism, agreeableness, conscientiousness, and openness to experience. Extraversion is characterised by traits such as talkativeness, assertiveness, sociability, and activity. Neuroticism pertains to emotional instability, anxiety, psychopathy, and insecurity. Agreeableness encompasses qualities such as being good-natured, friendly, forgiving, and tolerant. Conscientiousness relates to being careful, diligent, and organised. Lastly,

openness to experience encompasses traits such as imagination, curiosity, artistic inclination, and creativity (Resett et al. 2017; Geel et al. 2017). Based on empirical investigations utilising the Big Five personality traits, it has been observed that agreeableness and conscientiousness exhibit a negative correlation with cyberbullying, whereas extraversion and neuroticism demonstrate a positive correlation (Festl et al. 2013; Geel et al. 2017). Numerous studies have examined social network data derived from Facebook posts, tweets, and text messages in order to ascertain the potential associations between the Big Five personality traits and other characteristics and language usage (Balakrishnan et al. 2019; Holtgraves 2011; Kern et al. 2013; Sumner et al. 2012).

To predict the personality type of the user in the cyberbullying case study dataset, 100 most common words used in social media provided by (Schwartz et al. 2013) were used which are positively or negatively correlated with each of the personality types (agreeableness, conscientiousness, extraversion, openness, and neuroticism).

## **4.4 Feature Space Construction and Predictive Feature Integration**

The feature space used in this chapter combines STREAM-derived semantic features with a set of predictive cues motivated by prior cyberbullying research. STREAM features are constructed using semantic orientation scores derived from pointwise mutual information statistics, capturing the association between terms and severity-linked semantic poles. These features encode not only the presence of words but their contextual alignment with harmful or neutral orientations.

In addition to semantic orientation, linguistic and behavioural indicators are incorporated. These include markers related to writing style, emotional expression, and interaction patterns that have been shown to correlate with abusive behaviour in prior studies. Demographic proxies and personality-related cues are also encoded

where derivable from the text.

Importantly, predictive features are not treated as auxiliary inputs or evaluated through ablation studies. Instead, they are fully integrated into the primary feature space and processed identically to STREAM-derived features throughout feature selection and model training. This design reflects the underlying philosophy of STREAM: semantic interpretation in real-world moderation does not occur in isolation from contextual cues, and artificially separating these signals would misrepresent how such information is used in practice.

## 4.5 Experimental Setup and Emergency-Mode Evaluation

All experiments reported in this chapter follow the unified performance evaluation framework described in Chapter 3. The task is treated exclusively as a four-class classification problem, with no binary reformulation or label merging.

Due to the computational cost associated with STREAM feature construction and semantic scoring, all Cyberbullying experiments were executed using the emergency-mode configuration. Emergency mode preserves the core methodological principles of the evaluation framework, including fold-safe processing, pipeline isolation, and nested model selection, while reducing the number of outer and inner folds to ensure feasible execution. This reduction affects computational redundancy rather than evaluation validity and is applied consistently wherever emergency mode is required across the thesis.

All preprocessing, feature extraction, feature selection, and model fitting steps are performed strictly within the training partitions of each fold. This ensures that no information leakage occurs between training and evaluation data, a concern that is particularly acute in high-dimensional text classification pipelines.

## 4.6 Results and Analysis

This section analyses the Cyberbullying Severity results under the emergency-mode evaluation framework described in Chapter 3. The discussion is structured to make the methodological implications explicit, since this is the first empirical chapter following the STREAM formulation. All results treat the dataset strictly as a four-class problem (*none*, *low*, *medium*, *high*), and all predictive features are integrated into the main feature space rather than handled as auxiliary signals.

### 4.6.1 Class distribution and imbalance implications

Table 4.1 summarises the class distribution. The imbalance ratio increases sharply as severity rises, with *high* being the sparsest class. This distribution has two direct consequences for interpretation. First, accuracy may remain comparatively stable even when minority classes are poorly modelled, because correct prediction of *none* dominates the objective. Second, weighted  $F_1$  becomes the more diagnostic summary metric, as it penalises systematic minority-class collapse more strongly than accuracy.

Table 4.1: Class distribution and imbalance ratios

| Class  | Full Dataset | Training Set | Test Set | Imbalance Ratio |
|--------|--------------|--------------|----------|-----------------|
| None   | 20598        | 16478        | 4120     | 1.00            |
| Low    | 12736        | 10189        | 2547     | 1.62            |
| Medium | 4875         | 3900         | 975      | 4.23            |
| High   | 2101         | 1681         | 420      | 9.80            |

### 4.6.2 Emergency-mode nested cross-validation results

The primary emergency-mode comparison is reported in Table 4.2. The table is intentionally exhaustive, covering all evaluated combinations of representation (BoW, Word2Vec, STREAM) and classifier (NB, KNN, DT, RF, SVM), because this dataset is used to establish the empirical baseline for STREAM before later chapters focus on generalisation.

Two patterns are important. First, the strongest nested-CV weighted  $F_1$  values are obtained by BoW with tree-based ensembles (RF) and decision trees (DT), followed closely by BoW+SVM. In contrast, the best-performing STREAM configuration under nested CV is STREAM+RF, followed by STREAM+SVM and STREAM+NB. This ordering indicates that, under the reduced nested CV setting, improvements from semantic orientation are not uniformly dominant across all learners. Second, the Kappa values remain relatively modest even for the best-performing configurations, signalling that agreement beyond chance is constrained by label ambiguity and imbalance. This is consistent with the severity formulation, where the boundary between *low* and *medium* is often subtle and the *high* class is sparse.

A critical point for this chapter is that STREAM should not be framed as universally superior on this dataset under every evaluation scenario. Instead, the results show that, under nested CV, strong lexical baselines remain competitive, and STREAM’s advantage must be examined in conjunction with the stress tests and confusion patterns rather than asserted from a single aggregate score.

Table 4.2: Comparative performance summary across approaches and classifiers (emergency-mode nested CV).

| Approach | Classifier | F1_Mean      | F1_Std       | Accuracy     | Kappa        |
|----------|------------|--------------|--------------|--------------|--------------|
| BoW      | RF         | <b>0.782</b> | 0.001        | 0.818        | <b>0.301</b> |
| BoW      | DT         | 0.774        | 0.003        | 0.817        | 0.259        |
| BoW      | SVM        | 0.770        | 0.002        | <b>0.819</b> | 0.243        |
| STREAM   | RF         | 0.761        | 0.002        | 0.793        | 0.248        |
| BoW      | KNN        | 0.760        | 0.006        | 0.810        | 0.207        |
| STREAM   | NB         | 0.757        | 0.001        | 0.782        | 0.242        |
| Word2Vec | RF         | 0.757        | 0.005        | 0.815        | 0.191        |
| STREAM   | SVM        | 0.756        | 0.003        | 0.809        | 0.195        |
| STREAM   | DT         | 0.756        | 0.002        | 0.784        | 0.237        |
| Word2Vec | KNN        | 0.743        | 0.005        | 0.802        | 0.133        |
| STREAM   | KNN        | 0.739        | 0.003        | 0.776        | 0.156        |
| Word2Vec | DT         | 0.716        | 0.005        | 0.757        | 0.081        |
| Word2Vec | SVM        | 0.713        | <b>0.000</b> | 0.802        | 0.000        |
| Word2Vec | NB         | 0.261        | 0.008        | 0.211        | 0.039        |
| BoW      | NB         | 0.194        | 0.015        | 0.174        | 0.046        |

### 4.6.3 Holdout evaluation and confidence intervals

Holdout performance is reported in Table 4.3. The holdout table is exhaustive, and includes the nested-CV best  $F_1$  for each run as well as the 95% confidence interval bounds for holdout  $F_1$ . The strongest holdout  $F_1$  values are achieved by BoW-based models, with BoW+RF and BoW+DT at the top. STREAM-based models form the next tier (STREAM+RF, STREAM+DT, STREAM+NB, STREAM+SVM), with Word2Vec baselines generally below STREAM, except for Word2Vec+RF which is competitive in this particular split.

The key analytical observation is the stability of this ordering across related models. For example, BoW+RF and BoW+DT have similar holdout  $F_1$  (0.778 vs 0.773) and comparable confidence intervals, suggesting that the advantage is not driven by a single brittle configuration. Conversely, Word2Vec+SVM presents an instructive failure mode: although its accuracy is high (0.801), its  $F_1$  is notably lower (0.713) and Kappa is zero, indicating severe class-wise imbalance handling despite majority-class correctness. This is a concrete example of why weighted  $F_1$  and Kappa must be reported jointly for this dataset.

Table 4.3: Holdout test set performance

| Experiment   | Accuracy     | Precision    | Recall       | F1           | Kappa        | CV_F1_best   | F1_lo95      | F1_hi95      |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| BoW+RF       | 0.814        | 0.770        | 0.814        | <b>0.778</b> | <b>0.294</b> | <b>0.782</b> | <b>0.756</b> | <b>0.801</b> |
| BoW+DT       | 0.808        | 0.768        | 0.808        | 0.773        | 0.284        | 0.778        | 0.751        | 0.796        |
| BoW+SVM      | <b>0.815</b> | <b>0.793</b> | <b>0.815</b> | 0.768        | 0.240        | 0.780        | 0.746        | 0.791        |
| STREAM+RF    | 0.793        | 0.739        | 0.793        | 0.758        | 0.235        | 0.766        | 0.736        | 0.779        |
| Word2Vec+RF  | <b>0.815</b> | 0.752        | <b>0.815</b> | 0.756        | 0.193        | 0.748        | 0.731        | 0.778        |
| STREAM+DT    | 0.788        | 0.734        | 0.788        | 0.753        | 0.214        | 0.759        | 0.731        | 0.774        |
| STREAM+NB    | 0.780        | 0.734        | 0.780        | 0.752        | 0.224        | 0.759        | 0.731        | 0.774        |
| BoW+KNN      | 0.808        | 0.758        | 0.808        | 0.751        | 0.167        | 0.767        | 0.729        | 0.776        |
| STREAM+SVM   | 0.806        | 0.742        | 0.806        | 0.751        | 0.175        | 0.765        | 0.729        | 0.773        |
| Word2Vec+KNN | 0.802        | 0.732        | 0.802        | 0.739        | 0.114        | 0.741        | 0.715        | 0.759        |
| STREAM+KNN   | 0.772        | 0.719        | 0.772        | 0.738        | 0.165        | 0.742        | 0.717        | 0.761        |
| Word2Vec+DT  | 0.770        | 0.700        | 0.770        | 0.726        | 0.114        | 0.721        | 0.702        | 0.748        |
| Word2Vec+SVM | 0.801        | 0.642        | 0.801        | 0.713        | 0.000        | 0.730        | 0.689        | 0.735        |
| Word2Vec+NB  | 0.209        | 0.747        | 0.209        | 0.251        | 0.047        | 0.337        | 0.230        | 0.274        |
| BoW+NB       | 0.186        | 0.782        | 0.186        | 0.216        | 0.048        | 0.404        | 0.195        | 0.237        |

#### 4.6.4 Ten-fold cross-validation results

Table 4.4 reports a standard ten-fold cross-validation evaluation for reference. These results are not used for thesis-level inference, because the authoritative evaluation protocol is the fold-safe emergency-mode nested framework. However, the ten-fold results are useful for demonstrating that the same broad hierarchy of methods persists: BoW+RF remains strong, STREAM+RF and STREAM+SVM remain competitive, and Word2Vec-based models show mixed behaviour depending on classifier.

The most cautionary result remains Word2Vec+SVM, where accuracy is high but Kappa is effectively zero, indicating that the classifier can default to majority structures while appearing strong under accuracy-driven reporting. This reinforces the methodological necessity of the weighted- $F_1$ /Kappa reporting combination used throughout the thesis.

Table 4.4: Comprehensive 10-fold CV metrics

| Experiment   | F1           | Accuracy     | Precision    | Recall       | Kappa        |
|--------------|--------------|--------------|--------------|--------------|--------------|
| BoW+RF       | <b>0.785</b> | 0.819        | 0.778        | 0.819        | <b>0.314</b> |
| BoW+SVM      | 0.779        | <b>0.827</b> | <b>0.806</b> | <b>0.827</b> | 0.265        |
| STREAM+RF    | 0.767        | 0.796        | 0.751        | 0.796        | 0.267        |
| Word2Vec+RF  | 0.762        | 0.816        | 0.761        | 0.816        | 0.216        |
| STREAM+SVM   | 0.761        | 0.810        | 0.756        | 0.810        | 0.215        |
| STREAM+NB    | 0.760        | 0.784        | 0.744        | 0.784        | 0.252        |
| BoW+DT       | 0.756        | 0.758        | 0.755        | 0.758        | 0.287        |
| STREAM+DT    | 0.737        | 0.741        | 0.733        | 0.741        | 0.226        |
| BoW+KNN      | 0.736        | 0.755        | 0.720        | 0.755        | 0.190        |
| STREAM+KNN   | 0.722        | 0.723        | 0.721        | 0.723        | 0.190        |
| Word2Vec+KNN | 0.722        | 0.720        | 0.725        | 0.720        | 0.198        |
| Word2Vec+SVM | 0.713        | 0.802        | 0.642        | 0.802        | 0.000        |
| Word2Vec+DT  | 0.695        | 0.684        | 0.708        | 0.684        | 0.143        |
| Word2Vec+NB  | 0.276        | 0.224        | 0.746        | 0.224        | 0.049        |
| BoW+NB       | 0.201        | 0.178        | 0.789        | 0.178        | 0.047        |

#### 4.6.5 Stress tests: keep=0.6 and keep=0.4

Tables 4.5 and 4.6 report the comprehensive stress-test metrics under 60% and 40% training-data retention. These tables are exhaustive across representations and clas-

sifiers, and they should be read as stability diagnostics rather than as independent model-selection outcomes.

Under keep=0.6, the relative ordering remains broadly similar to the main evaluation, with BoW tree-based models (DT/RF) remaining strongest in mean  $F_1$ . STREAM models remain competitive but do not dominate. Under keep=0.4, the top of the table still belongs to BoW-based DT/RF/SVM variants, while STREAM variants cluster in the mid-range.

The more informative finding is not the identity of the single best method, but the way different modelling families degrade. The Naive Bayes baselines (particularly BoW+NB and Word2Vec+NB) show pathological behaviour under both stress settings, with very low  $F_1$  despite non-trivial precision. This is consistent with a failure mode where a model predicts a small subset of classes confidently while ignoring others, producing superficially reasonable precision but poor recall and poor overall  $F_1$ . In contrast, the leading DT/RF configurations show relatively small standard deviations across folds, suggesting that their performance is not highly sensitive to the reduced training size under emergency-mode folds.

Table 4.5: Comprehensive metrics under imbalance stress (keep=0.6)

| Experiment   | F1 Mean      | F1 Std       | Accuracy Mean | Accuracy Std | Precision Mean | Precision Std | Recall Mean  | Recall Std   | Kappa Mean   | Kappa Std    |
|--------------|--------------|--------------|---------------|--------------|----------------|---------------|--------------|--------------|--------------|--------------|
| BoW+DT       | <b>0.778</b> | 0.001        | 0.814         | 0.001        | 0.775          | 0.001         | 0.814        | 0.001        | <b>0.288</b> | 0.003        |
| BoW+RF       | 0.774        | 0.003        | 0.815         | 0.002        | 0.772          | 0.001         | 0.815        | 0.002        | 0.266        | 0.016        |
| BoW+SVM      | 0.767        | 0.004        | <b>0.816</b>  | 0.002        | <b>0.789</b>   | 0.010         | <b>0.816</b> | 0.002        | 0.234        | 0.021        |
| STREAM+RF    | 0.758        | 0.001        | 0.788         | 0.002        | 0.741          | 0.002         | 0.788        | 0.002        | 0.235        | 0.003        |
| BoW+KNN      | 0.754        | 0.005        | 0.808         | 0.003        | 0.761          | 0.003         | 0.808        | 0.003        | 0.178        | 0.022        |
| STREAM+DT    | 0.751        | 0.007        | 0.777         | 0.008        | 0.734          | 0.008         | 0.777        | 0.008        | 0.223        | 0.023        |
| STREAM+SVM   | 0.751        | 0.003        | 0.806         | 0.002        | 0.745          | 0.001         | 0.806        | 0.002        | 0.172        | 0.015        |
| STREAM+NB    | 0.750        | 0.003        | 0.777         | 0.006        | 0.734          | 0.004         | 0.777        | 0.006        | 0.218        | 0.015        |
| Word2Vec+KNN | 0.740        | 0.007        | 0.804         | 0.004        | 0.737          | 0.012         | 0.804        | 0.004        | 0.121        | 0.024        |
| Word2Vec+RF  | 0.736        | 0.006        | 0.801         | 0.010        | 0.737          | 0.012         | 0.801        | 0.010        | 0.114        | 0.016        |
| STREAM+KNN   | 0.732        | 0.001        | 0.768         | <b>0.000</b> | 0.711          | 0.001         | 0.768        | <b>0.000</b> | 0.135        | 0.005        |
| Word2Vec+SVM | 0.713        | <b>0.000</b> | 0.802         | <b>0.000</b> | 0.642          | <b>0.000</b>  | 0.802        | <b>0.000</b> | 0.000        | <b>0.000</b> |
| Word2Vec+DT  | 0.709        | 0.002        | 0.757         | 0.003        | 0.674          | 0.002         | 0.757        | 0.003        | 0.052        | 0.011        |
| Word2Vec+NB  | 0.230        | 0.056        | 0.189         | 0.039        | 0.704          | 0.014         | 0.189        | 0.039        | 0.023        | 0.012        |
| BoW+NB       | 0.198        | 0.013        | 0.176         | 0.007        | 0.777          | 0.009         | 0.176        | 0.007        | 0.045        | 0.002        |

Table 4.6: Comprehensive metrics under imbalance stress (keep=0.4)

| Experiment   | F1 Mean | F1 Std       | Accuracy Mean | Accuracy Std | Precision Mean | Precision Std | Recall Mean  | Recall Std   | Kappa Mean | Kappa Std    |
|--------------|---------|--------------|---------------|--------------|----------------|---------------|--------------|--------------|------------|--------------|
| BoW+DT       | 0.772   | 0.003        | 0.810         | 0.002        | 0.765          | 0.005         | 0.810        | 0.002        | 0.271      | 0.016        |
| BoW+RF       | 0.771   | 0.001        | 0.811         | 0.006        | 0.767          | 0.006         | 0.811        | 0.006        | 0.258      | 0.004        |
| BoW+SVM      | 0.767   | 0.008        | <b>0.812</b>  | 0.012        | <b>0.782</b>   | 0.029         | <b>0.812</b> | 0.012        | 0.236      | 0.020        |
| STREAM+NB    | 0.753   | 0.002        | 0.779         | 0.003        | 0.737          | 0.003         | 0.779        | 0.003        | 0.229      | 0.008        |
| STREAM+RF    | 0.753   | 0.005        | 0.786         | 0.003        | 0.736          | 0.007         | 0.786        | 0.003        | 0.211      | 0.019        |
| BoW+KNN      | 0.749   | 0.003        | 0.808         | 0.002        | 0.760          | 0.009         | 0.808        | 0.002        | 0.153      | 0.012        |
| STREAM+SVM   | 0.748   | 0.002        | 0.804         | 0.005        | 0.737          | 0.002         | 0.804        | 0.005        | 0.161      | 0.008        |
| STREAM+DT    | 0.741   | 0.005        | 0.772         | 0.004        | 0.722          | 0.006         | 0.772        | 0.004        | 0.183      | 0.019        |
| Word2Vec+KNN | 0.734   | 0.004        | 0.802         | 0.002        | 0.733          | 0.012         | 0.802        | 0.002        | 0.094      | 0.021        |
| Word2Vec+RF  | 0.733   | 0.006        | 0.803         | 0.007        | 0.742          | 0.015         | 0.803        | 0.007        | 0.092      | 0.017        |
| STREAM+KNN   | 0.730   | 0.005        | 0.763         | 0.005        | 0.709          | 0.007         | 0.763        | 0.005        | 0.140      | 0.024        |
| Word2Vec+SVM | 0.713   | <b>0.000</b> | 0.802         | <b>0.000</b> | 0.642          | <b>0.000</b>  | 0.802        | <b>0.000</b> | 0.000      | <b>0.000</b> |
| Word2Vec+DT  | 0.696   | 0.006        | 0.726         | 0.011        | 0.671          | 0.006         | 0.726        | 0.011        | 0.046      | 0.021        |
| BoW+NB       | 0.270   | 0.018        | 0.223         | 0.013        | 0.748          | 0.004         | 0.223        | 0.013        | 0.051      | 0.005        |
| Word2Vec+NB  | 0.225   | 0.037        | 0.186         | 0.024        | 0.704          | 0.012         | 0.186        | 0.024        | 0.022      | 0.003        |

#### 4.6.6 Scenario-level stress summaries

To complement the fold-aggregated stress tables, Tables 4.7 and 4.8 summarise the same stress scenarios in a compact form across models. These summaries are useful for cross-checking that performance degradation trends are consistent when viewed through the simplified scenario reporting interface, and for observing that the overall ranking is not dramatically re-ordered by the stress manipulations.

Table 4.7: stress60: Performance metrics across different evaluation scenarios

| Scenario     | Accuracy     | Precision    | Recall       | F1           | Kappa        |
|--------------|--------------|--------------|--------------|--------------|--------------|
| STREAM+NB    | 0.776        | 0.736        | 0.776        | 0.752        | 0.231        |
| STREAM+KNN   | 0.768        | 0.710        | 0.768        | 0.731        | 0.138        |
| STREAM+SVM   | 0.800        | 0.733        | 0.800        | 0.744        | 0.148        |
| STREAM+DT    | 0.777        | 0.730        | 0.777        | 0.748        | 0.208        |
| STREAM+RF    | 0.781        | 0.727        | 0.781        | 0.748        | 0.204        |
| BoW+NB       | 0.148        | 0.777        | 0.148        | 0.157        | 0.033        |
| BoW+KNN      | 0.809        | 0.768        | 0.809        | 0.754        | 0.179        |
| BoW+SVM      | <b>0.815</b> | <b>0.795</b> | <b>0.815</b> | 0.769        | 0.250        |
| BoW+DT       | 0.813        | 0.767        | 0.813        | 0.763        | 0.213        |
| BoW+RF       | <b>0.815</b> | 0.774        | <b>0.815</b> | <b>0.777</b> | <b>0.284</b> |
| Word2Vec+NB  | 0.209        | 0.737        | 0.209        | 0.262        | 0.037        |
| Word2Vec+KNN | 0.803        | 0.733        | 0.803        | 0.736        | 0.103        |
| Word2Vec+SVM | 0.801        | 0.642        | 0.801        | 0.713        | 0.000        |
| Word2Vec+DT  | 0.745        | 0.688        | 0.745        | 0.711        | 0.081        |
| Word2Vec+RF  | 0.812        | 0.748        | 0.812        | 0.745        | 0.144        |

Table 4.8: stress40: Performance metrics across different evaluation scenarios

| Scenario     | Accuracy     | Precision    | Recall       | F1           | Kappa        |
|--------------|--------------|--------------|--------------|--------------|--------------|
| STREAM+NB    | 0.776        | 0.728        | 0.776        | 0.747        | 0.205        |
| STREAM+KNN   | 0.773        | 0.714        | 0.773        | 0.734        | 0.139        |
| STREAM+SVM   | 0.797        | 0.733        | 0.797        | 0.750        | 0.181        |
| STREAM+DT    | 0.779        | 0.730        | 0.779        | 0.748        | 0.203        |
| STREAM+RF    | 0.789        | 0.738        | 0.789        | 0.755        | 0.221        |
| BoW+NB       | 0.220        | 0.767        | 0.220        | 0.266        | 0.056        |
| BoW+KNN      | 0.804        | 0.757        | 0.804        | 0.742        | 0.127        |
| BoW+SVM      | 0.809        | 0.768        | 0.809        | 0.760        | 0.213        |
| BoW+DT       | 0.811        | 0.776        | 0.811        | 0.774        | <b>0.288</b> |
| BoW+RF       | <b>0.816</b> | <b>0.777</b> | <b>0.816</b> | <b>0.779</b> | 0.284        |
| Word2Vec+NB  | 0.219        | 0.731        | 0.219        | 0.277        | 0.036        |
| Word2Vec+KNN | 0.804        | 0.733        | 0.804        | 0.739        | 0.115        |
| Word2Vec+SVM | 0.801        | 0.642        | 0.801        | 0.713        | 0.000        |
| Word2Vec+DT  | 0.710        | 0.669        | 0.710        | 0.688        | 0.037        |
| Word2Vec+RF  | 0.809        | 0.751        | 0.809        | 0.741        | 0.124        |

#### 4.6.7 Hypothesis tests

Table 4.9 summarises the hypothesis tests comparing the best STREAM model against the best non-STREAM model across scenarios. The nested-CV comparison does not detect a statistically significant difference at the reported threshold, despite a large effect size. In contrast, the holdout comparison favours the non-STREAM baseline (BoW+RF) with a statistically significant difference and a large effect size. This pattern is consistent with the earlier observation that, on this dataset, strong lexical baselines remain competitive and in some cases superior, and it reinforces the need to interpret STREAM performance in terms of measured behaviour under stress and misclassification patterns rather than relying on a single aggregate score.

Table 4.9: Summary of hypothesis tests across evaluation scenarios

| Scenario  | Best STREAM | Best Non-STREAM | p-value / Outcome                 |
|-----------|-------------|-----------------|-----------------------------------|
| Nested CV | STREAM + RF | BoW + RF        | 0.250 / No significant difference |
| Holdout   | STREAM + RF | BoW + RF        | 0.000 / BoW+RF superior           |

#### 4.6.8 Confusion matrices and class-wise error structure

Aggregate metrics conceal the mechanisms of failure under severe imbalance, so confusion matrices are essential for interpreting the severity task. Figure 4.2 shows the holdout confusion matrices for the leading BoW configurations. The dominant error mode is minority-class absorption into the *none* class. For example, in BoW+RF, a large fraction of *medium* instances are predicted as *none* (116 versus 24 correct), and *high* instances are also heavily mapped to *none* (93 versus 24 correct). The same structural pattern appears in BoW+DT and BoW+SVM, with relatively small counts in the *high* diagonal cell compared to the off-diagonal *high*→*none* errors. This pattern is consistent with the difficulty of learning reliable decision boundaries for sparse severe classes when the majority class is both frequent and lexically heterogeneous.

Figure 4.3 extends this analysis to the keep=0.6 stress scenario across all representations and classifiers. Two behaviours are particularly informative. First, the collapse of Naive Bayes baselines is visible as extremely skewed prediction distributions, indicating that these models are not maintaining meaningful multi-class separation under reduced supervision. Second, even for stronger models, the adjacent-class confusions remain secondary to the dominant *medium/high*→*none* absorption, showing that the principal difficulty is not only ambiguity between adjacent severity categories but also the scarcity and lexical diversity of minority classes relative to the majority class.

#### 4.6.9 Critical synthesis

Taken together, the Cyberbullying Severity results establish three dataset-specific lessons that inform the rest of the thesis.

First, strong lexical baselines remain competitive under emergency-mode evaluation, and the best-performing model in this dataset is not uniformly STREAM-based across scenarios. This is not an anomaly to be explained away. Cyberbullying

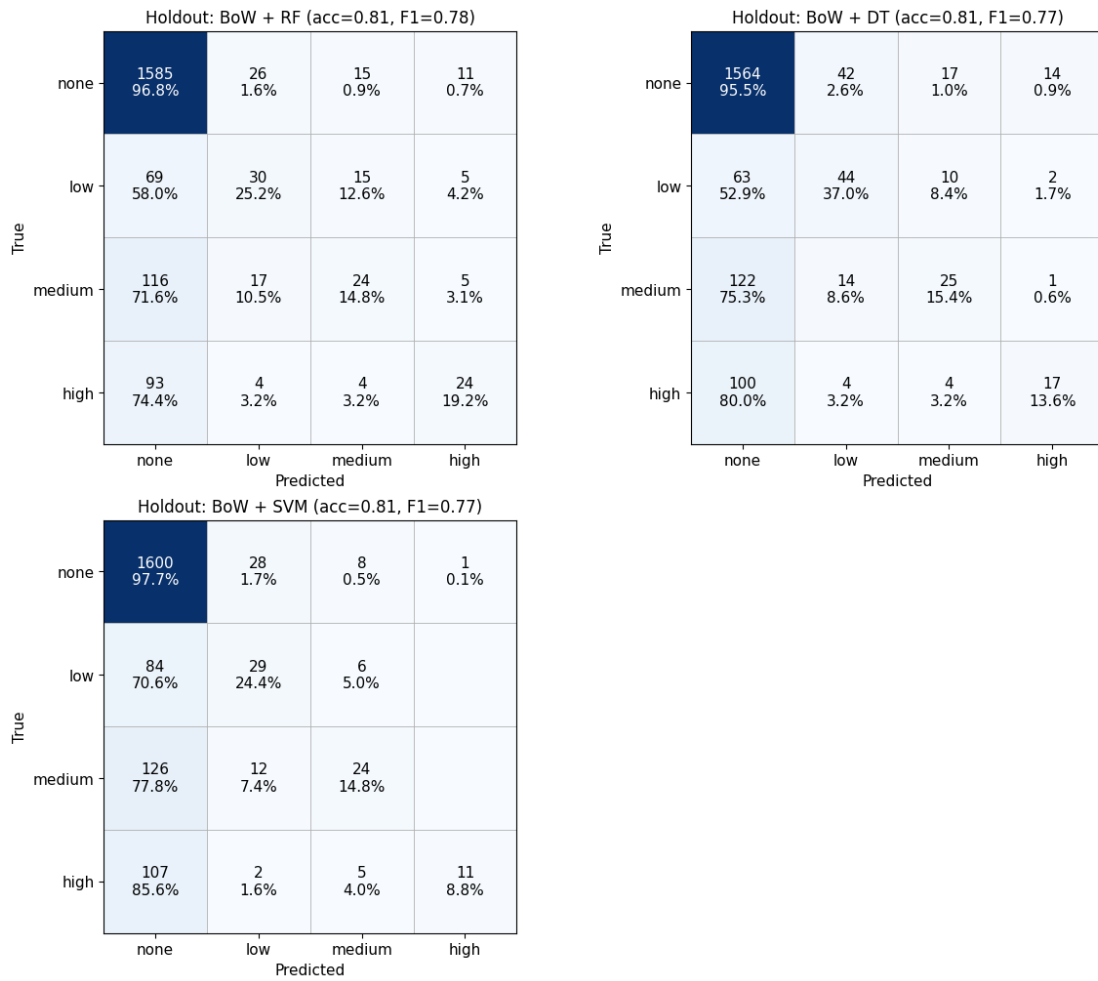


Figure 4.2: Cyberbullying Severity: Holdout confusion matrices for leading BoW-based models.



Figure 4.3: Cyberbullying Severity: Confusion matrices under keep=0.6 stress testing across all approaches and classifiers.

severity is strongly correlated with explicit lexical cues in a subset of cases, and tree-based lexical models can exploit these cues effectively even under imbalance.

Second, the evaluation demonstrates why the thesis insists on fold-safe pipelines and multi-metric reporting. Several configurations achieve high accuracy while exhibiting weak  $F_1$  and negligible Kappa, indicating that they can appear effective under accuracy-only reporting while failing to separate minority severity classes. This would be particularly misleading for moderation systems, where the cost of missing *medium/high* severity content is materially different from misclassifying mild content.

Third, stress tests and confusion matrices reveal that the principal difficulty in this dataset is minority-class modelling under sparse supervision. Under both  $\text{keep}=0.6$  and  $\text{keep}=0.4$ , the dominant degradation mode is not random noise but structured collapse of minority classes into *none*. The presence of this pattern even among high-performing baselines motivates the broader thesis focus: improvements should be assessed not only by aggregate scores but by whether the representation and learner combination maintains separability of the minority classes under realistic constraints.

This chapter therefore provides an empirically grounded baseline for STREAM, showing both where semantic orientation contributes and where the dataset’s lexical structure allows strong baselines to compete. Subsequent chapters use this foundation to evaluate whether STREAM’s representational assumptions generalise across domains where lexical shortcuts are weaker, class structure differs, and semantic variability is broader.

## 4.7 Domain Analysis - Cyberbullying

The case study aimed to improve automated cyberbullying detection, which is the main step towards automated systems for analyzing contemporary social online behaviours that may adversely affect mental health. Through execution of the case study a step forward has been taken and limitations in current cyberbullying detec-

tion systems has been identified. The primary objective and contribution of the case study was to establish a systematic approach (machine learning multi-classifiers) for assessing severity level in cyberbullying behavioural text using multi-class classification. To test the significance of the proposed STREAM method for detecting cyberbullying severity, a publicly available harassment dataset was examined. A comprehensive series of experiments was conducted for detecting cyberbullying severity (in tweets) in binary scheme (either cyberbullying behaviour exists in the tweet or not) and multi-classification scheme (low, medium, high, or none).

It was found that specific features contribute to cyberbullying severity detection. When cyberbullying behavior is detected then:

1. What type of personality traits fall under different severity levels (Low, Medium, or High)?
2. What is their gender?
3. What age group do they belong to?
4. What time do they tweet post?
5. Since when has the user been using Twitter?
6. What makes their tweet cyberbullied?

To identify these features, multinomial logistic regression was conducted to extract the odd ratio of all features (Content, Network, Activity, Personality, and User) that are applied in the model Figure 4.4. With the tree map plot, both the size of the rectangles and their colour are determined by the odds ratio of each feature—the higher the odds ratio for each feature, the darker and larger its box.

It was found that the likelihood of users being classified in low severity level category of cyberbullying are users with neuroticism behaviour, age group between 19-22 years, tweet time mostly at midnight, their Twitter account is relatively new (< 1 year). Features with the highest odds ratio Figure 4.4 for detecting low-level severity include: age group between 19-22 years and users since < 1 year of account.



Figure 4.4: A treemap with rectangle sizes and colors determined by each feature's odds ratio

Cyberbullying tweets categorized as medium severity belong to politics and racism. Unlike low severity category, medium severity people have the personality trait of agreeableness, age group between 23-29 years, tweet time mostly in afternoon and midnight, their Twitter account between 1 to 2 years. In this group, the rate of perpetrator's tweet being favored by other users is a bit higher compared to low-level severity users. The highest odd ratio Figure 4.4 for detecting medium severity level predictors are neuroticism, age group between 23-29 years, and Twitter user between 1 to 2 years.

Sexual and appearance-related tweets are categorized under a high-level severity category of cyberbullying. The likelihood of users to be categorized with a high-level of cyberbullying severity are Figure 4.4; neuroticism and extraversion, their Twitter account age between 2 to 5 years. In this group, perpetrator tweet is favored by other users is higher than low and medium-level cyberbullied severity users. Predictors with the highest odds ratio Figure 4.4 for detecting high cyberbullying severity levels are neuroticism and extraversion, and the number of times tweet has been favored. The relation of neuroticism and extraversion to cyberbullying perpetration was also reported by (Mitsopoulou et al. 2015) and (Festl et al. 2013). Al-

though gender is a good feature to know about the user (and is correlated with other features such as network, activity, and user to detect cyberbullying tweet), our finding suggests otherwise. In any of the cyberbullying severity categories Figure 4.4, gender does not play an important role when identifying the level of severity in the cyberbullying content.

In the case study, predicted features such as predicting gender and user's age were extracted from vocabulary, demonstrating a notable enhancement in the accuracy of overall classifiers. The content of aforementioned used vocabulary should however be further studied and extended for updated words and phrases. The algorithmic biasness due to this vocabulary could possibly generate systematic and repeatable errors that could eventually create unfair outcomes, such as privileging one arbitrary group of users over others (F. Sun et al. 2019; R. Zhao et al. 2017).

The overall findings indicate that multiple features can be used to enhance cyberbullying detection, and choosing specific features is a crucial part in the process of cyberbullying detection. Most of the previous cyberbullying related studies (Algaradi et al. 2016; Balakrishnan et al. 2019; Chatzakou, Vakali, et al. 2017), focused on feature selection rather than how text is represented at a pre-processing stage. The cyberbullying case study has shed some light on the text representation at the pre-processing stage before important features are fed into the desired machine learning model for training. Classifiers' performance for the proposed vector-based model, indicating the proposed approach performed best in detecting cyberbullying severity in online social networks (accuracy of 93%) compared to baseline approaches. The findings are in line with previous empirical evidence that has shown the personality feature to be linked with cyberbullying behaviour (Algaradi et al. 2016; Alonso et al. 2017; Festl et al. 2013; Geel et al. 2017). Direct comparison is unfeasible with cyberbullying studies due to several reasons: the nature of availability of twitter dataset, scope of the study, and analysis mechanism. To test the significance of proposed approach, however, we tested the same dataset against two well-known text representation approaches, namely, BoW and Word2Vec.

## 4.8 Chapter Summary

This chapter evaluated multi-class cyberbullying severity detection on a Twitter-derived dataset, treating the task strictly as four-way classification over *none*, *low*, *medium*, and *high*. The chapter’s primary role within the thesis is to provide the first empirical grounding of STREAM under the final, fold-safe evaluation framework. Accordingly, all experiments were conducted using the pipeline-based protocol described in Chapter 5, executed in emergency mode to preserve methodological safeguards while remaining computationally feasible. Performance was reported using accuracy and weighted  $F_1$  as the primary summary measures, supported by agreement statistics and class-wise evidence where appropriate.

Baseline approaches, Bag-of-Words and Word2Vec, were evaluated across a common family of classifiers (NB, KNN, SVM, DT, RF) under identical evaluation conditions. These baselines provide an essential reference point for understanding how far conventional lexical and distributional representations can progress in an imbalanced, fine-grained severity setting, where accuracy alone can be dominated by majority classes. In parallel, the STREAM representation was evaluated under the same classifier suite. Predictive features motivated by prior cyberbullying literature, including demographic and behavioural cues, were fully integrated into the main feature space and treated as first-class features rather than auxiliary inputs, ensuring that all comparisons reflect consistent pipelines and fold-safe processing.

Across evaluation scenarios, the results show that strong lexical baselines can remain competitive on this dataset, while STREAM provides a semantically enriched representation that is particularly valuable when class boundaries are subtle and supervision is reduced. The mandatory stress tests ( $\text{keep}=0.6$  and  $\text{keep}=0.4$ ) extend the analysis beyond the base setting by amplifying minority-class scarcity. Under these perturbations, the chapter highlights how model families degrade, and whether relative ranking and class-wise behaviour persist. Confusion-matrix analysis is used to interpret the dominant error structure, with particular attention to adjacent-severity

confusions and the extent to which minority classes are absorbed into majority predictions under imbalance.

Overall, this chapter establishes a defensible empirical baseline for STREAM. Rather than relying on a single aggregate metric, it demonstrates how representation choice affects performance under fold-safe evaluation, under data reduction stress, and in terms of class-wise misclassification behaviour. This provides the methodological and empirical foundation for the subsequent chapters, which evaluate STREAM’s generalisation across datasets with different linguistic characteristics and class structures.

## 5 | Evaluating STREAM Across Diverse Textual Domains

### 5.1 Introduction

In the previous chapter, we demonstrated the effectiveness of the STREAM methodology on the cyberbullying dataset, a domain characterized by severe multi-class imbalance and a unique severity-level classification objective. To confirm that STREAM’s semantic feature engineering and imbalance mitigation strategies are broadly applicable, this chapter extends the evaluation to four additional datasets: Newspaper tweets (NY Times), Yelp reviews, Reuters, and 20 Newsgroups.

Before proceeding, it is important to highlight that the exact experimental settings and parameter selection vary slightly from those used in Chapter 4. For the cyberbullying dataset, we investigated feature selection approaches to compare their impact on performance. In this chapter, we streamline the methodology by concentrating on baseline representations (BoW and Word2Vec) and the STREAM approach, without extensive feature selection experiments. Furthermore, some datasets required particular handling—such as filtering out classes with extremely low sample counts to ensure stable stratified cross-validation—due to their class distributions. These differences illustrate the practical realities of text classification research, where each dataset’s inherent properties (class distribution, domain characteristics, and resource constraints) guide the methodological adjustments.

We also acknowledge that the experimental protocols vary slightly between preliminary tuning and final evaluation. For preliminary parameter selection (e.g., choosing  $K$  and distance metrics for KNN or kernels for SVM (subsection 5.2.2)), a lighter-weight cross-validation (e.g., 5-fold) may have been used to expedite the process. Final reported results, however, depend on more rigorous evaluation strategies (e.g., 10-fold cross-validation and a held-out test set) to ensure the highest standards of rigor.

Finally, due to computational considerations, a subset of 10,000 reviews from the multimillion-record Yelp dataset was used. Choosing representative subsets is a standard and recognized method in large-scale data research (Kowsari et al. 2019; Minaee et al. 2020), allowing for manageable experimentation and iteration without sacrificing the validity of comparative performance conclusions. Likewise, when exceedingly rare classes appeared in certain datasets, we exclude them to maintain meaningful class presence for stratified cross-validation. This step ensures stable metrics and comparisons, albeit at the cost of perfect fidelity to the dataset’s original class distribution.

These decisions, while introducing minor variations from previous chapters and across datasets, are standard and justified within the field. They confirm that each evaluation is both feasible and reliable, enabling a clear demonstration of STREAM’s adaptability and benefits.

### **5.1.1 Datasets and Their Characteristics**

#### **5.1.2 Newspaper (NY Times) Dataset**

The Newspaper dataset in this study is constructed from tweets sourced from the official Twitter profile<sup>1</sup> of the New York Times (NY Times). There are several reasons why the NY Times was chosen for this particular case study:

1. Reputability and Representativeness: The New York Times (NY) is widely ac-

---

<sup>1</sup><https://twitter.com/nytimes>

knowledgeed as one of the most influential and esteemed newspapers world-wide, renowned for its accurate, objective, and balanced reporting. Utilizing tweets from such a well-established media outlet ensures that the content reflects a wide range of news topics and editorial standards. In effect, the NY Times can be considered representative of newspaper journalism in general, offering a valuable dataset for research focused on understanding and categorizing the emotional tone of news-related tweets.

2. **Data Availability and Volume:** The NY Times Twitter account is exceptionally active, generating numerous tweets daily. For this study, tweets were extracted through Twitter’s API, collecting a maximum historical range allowed by the platform’s policies<sup>2</sup>. A total of 19,950 distinct tweets were collected from April 2019 to September 2019. This huge volume provided a sufficiently large and diverse corpus to facilitate meaningful emotion recognition experiments. The availability of this dataset and its rich textual content made it an excellent choice for our analysis.

### **Preprocessing and Feature Extraction:**

The tweets originally had minimal metadata: a unique tweet ID, the tweet’s text, and its timestamp. Pre-processing steps (as discussed in Section 4.3) included cleaning the text, removing irrelevant characters, and tokenizing the tweets. Similar feature extraction methods were then applied—creating BoW and Word2Vec representations for baseline comparisons, and employing the STREAM methodology (detailed in Section 3.2.5) for generating semantically enriched embeddings.

The distribution is significantly skewed (Figure 5.1): Passive dominates (6,267 samples), while classes such as Fun (587) and Unpleasant (383) are scarce. This multi-class imbalance scenario tests STREAM’s ability to capture subtle linguistic signals and improve minority class recognition in a domain unrelated to cyberbullying.

---

<sup>2</sup><https://developer.twitter.com/en/developer-terms/agreement-and-policy.html>

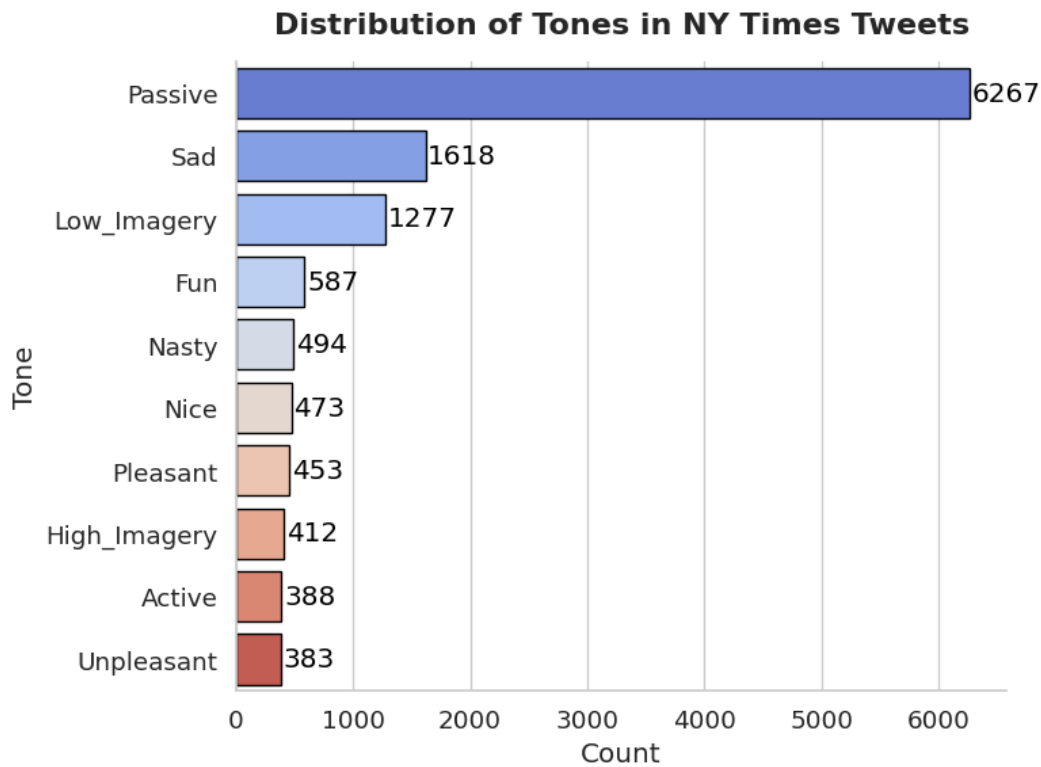


Figure 5.1: Class Distribution in NY Times Dataset

### Emotional Categorization via WDAL and RIOT Scan:

To analyze the emotional tone of the tweets, we employed the revised Dictionary of Affect in Language (WDAL) (Hoover 2015; Whissell 2009), which contains 8,742 words rated on dimensions including pleasantness, activation (activity), and imagination (imagery). Using the RIOT Scan software (Byod 2013), we mapped each cleaned tweet into one of several emotional categories derived from WDAL: *Active*, *Fun*, *High Imagery*, *Low Imagery*, *Nasty*, *Nice*, *Passive*, *Pleasant*, *Sad*, and *Unpleasant*.

Each word in WDAL provides a decimal rating that reflects how subjects perceive it along these affective dimensions. For example, the term “yesterday” is rated at 2.57 (pleasantness), 1.83 (activity), and 1.60 (imagery), indicating it is moderately pleasant, neutral-to-passive, and slightly abstract. By applying RIOT Scan, each tweet is designated an emotional label reflecting the dominant affective qualities of its lexical components.

Subsequent to this lexical analysis, untagged or inadequately rated tweets were

eliminated to guarantee that the final dataset comprised just of tweets with clear and reliable emotional annotations. After these refinements, the final dataset contained 12,352 tweets evenly distributed throughout the defined emotional categories. This meticulous curation ensures that the dataset meets the necessary characteristics for emotional analysis, providing a robust foundation for subsequent classification experiments.

### **Domain Considerations: Emotion vs. Sentiment Recognition:**

It is important to differentiate between emotion and sentiment detection in text. Although sentiment analysis (e.g., classifying text as “positive,” “negative,” or “neutral”) is a popular task NLP, emotion recognition is intrinsically more nuanced and complex. Emotions such as Happiness, Anger, or Love may correlated with sentiments like *Positive* or *Negative*, but the relationship is not absolutely one-to-one. Emotions may vary greatly based on personality, culture, context, and numerous other factors, making them more ambiguous and challenging to interpret.

The newspaper tweets dataset places prioritize the detection of emotional tones over mere polarity. Unlike many user-generated content scenarios where strong personal opinions dominate, the NY Times tweets are more formal and informational. Nonetheless, the variation in tone—ranging from *Sad* and *Nasty* to *Pleasant* and *Fun*—necessitates a nuanced approach to capture subtle lexical cues that reveal underlying affective states.

### **Summary of the Newspaper Dataset Setup:**

- Source: Official NY Times Twitter profile, chosen for its credibility, representativeness, and data availability.
- Data Volume and Range: 19,950 tweets collected from April to September 2019, filtered down to 12,352 emotionally annotated tweets.
- Preprocessing: Standard text cleaning and tokenization, followed by baseline

feature extractions (BoW, Word2Vec) and STREAM embeddings.

- Emotional Annotation: WDAL-based categories assigned via RIOT Scan, ensuring the final dataset offers distinct, reliably scored emotional classes.
- Domain Relevance: Emphasizes emotion recognition over simple sentiment detection, illustrating the complexity and subtlety of emotional tone classification in a respected news domain.

### 5.1.3 Yelp Reviews Dataset

The Yelp dataset comprises millions of reviews <https://www.yelp.com/dataset>. To maintain computational feasibility and facilitate repeated testing, we sampled 10,000 reviews, a practice commonly accepted in research for managing extensive datasets. After ensuring that each class (1.0 to 5.0 star ratings) included a sufficient number of samples, the class distribution remained skewed (Figure 5.2): class 5.0 (3,556 samples) is most frequent, while class 2.0 (611 samples) is the least frequent.

Predicting star ratings from review text is more complex than basic polarity categorization because it requires understanding subtle distinctions in user opinions and sentiments. By applying STREAM’s semantic orientation and enhanced embeddings, we aim to improve identification of these nuanced rating patterns, particularly for less common classes. Despite using only a subset of the larger dataset, this approach yields a realistic and sufficiently varied sample to test the effectiveness of STREAM in handling multi-class imbalance in opinion-rich content.

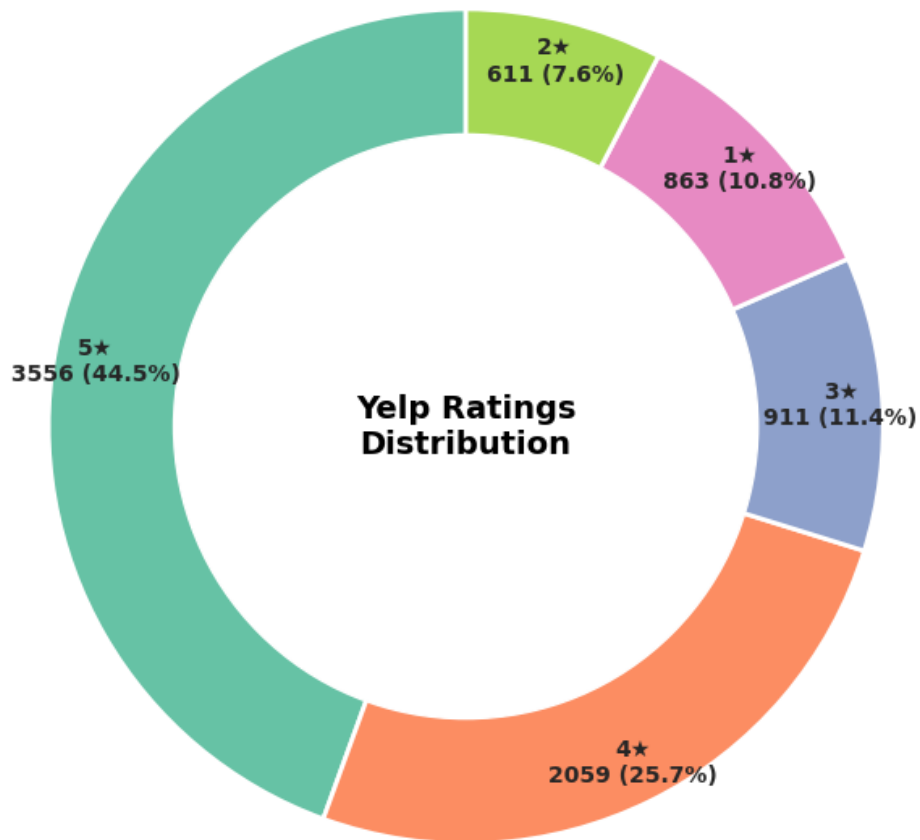


Figure 5.2: Class Distribution in Yelp Dataset

#### 5.1.4 Reuters Dataset

The Reuters-21578 corpus, accessed via the NLTK library's corpus downloader, serves as a common benchmark for text classification. The dataset was acquired using `nltk.download('reuters')` as described at [https://www.nltk.org/nltk\\_data/](https://www.nltk.org/nltk_data/), originally comprises documents categorized into multiple categories, often resulting in a multi-labeled classification scenario. However, handling multi-labeled data can complicate both training and evaluation, particularly when striving for stable stratified cross-validation. To manage complexity and ensure a straightforward multi-class, single-label environment, we refined the dataset to retain only documents with a single category assignment. Concretely, we selected those documents that possessed only one category label. This step not only simplifies the classification task but also facilitates clearer interpretation of results.

However, even after limiting the analysis to single-labeled documents, some categories remained extremely rare—sometimes with only one or two documents. Such sparse classes pose challenges for stratified cross-validation and can lead to unstable evaluation metrics. To resolve this, we applied an extra filtering criterion: classes appearing fewer than a minimal threshold (e.g., less than 2 documents) were removed. After these filtering steps, the resulting dataset contained 9,160 documents distributed over 56 unique topic categories.

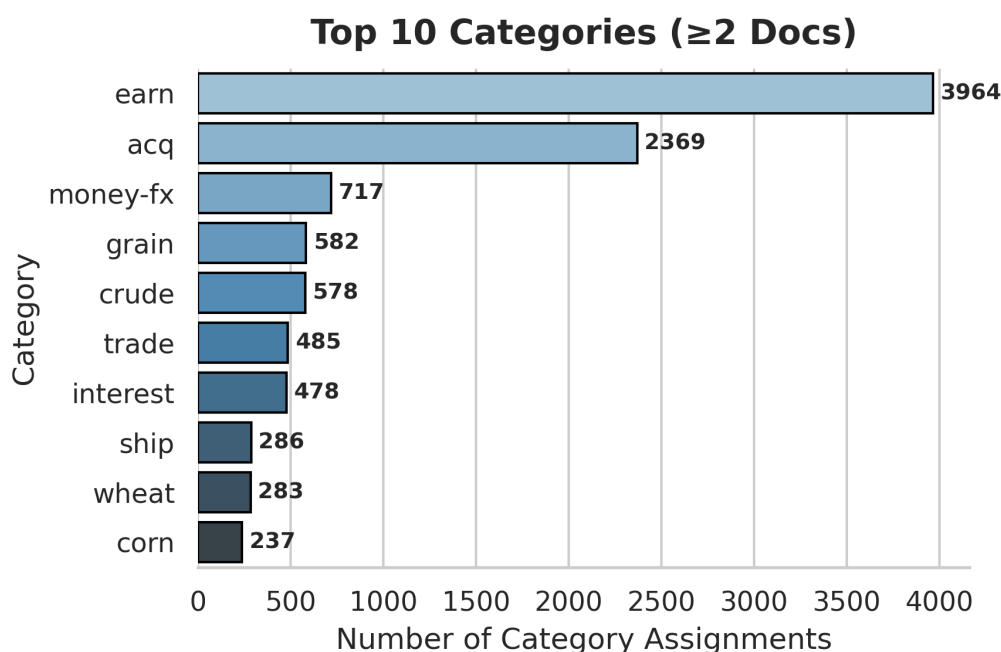


Figure 5.3: Top 10 Categories in Reuters Dataset

While this pre-processing ensures a more robust evaluation setup, it also change the original distribution of the Reuters corpus. Figure 5.3 highlights the top 10 categories, showing that a few dominant topics, such as earn and acq, comprise a large portion of the dataset, while many other categories have moderate or low frequencies. Thus, the final dataset still exhibits a long-tailed, imbalanced distribution, despite the filtering. Moreover, because we selected only single-labeled documents, the complexity of multi-label classification present in the original Reuters corpus is largely removed. As a result, the classification task here is exclusively single-label multi-class, which allows direct application of standard metrics (F1, Kappa) and

stratified cross-validation without additional multi-label complexity.

It is essential to note that these filtering decisions mean direct comparisons with previous studies on Reuters-21578 may require caution. Different researchers can apply varying subsets, filtering criteria, or include multi-labeled documents. Nevertheless, the resulting dataset is sufficiently robust and challenging enough to test the adaptability and effectiveness of the STREAM methodology in a professional, topic-rich environment.

### 5.1.5 20 Newsgroups Dataset

The 20 Newsgroups dataset is a well-established benchmark frequently used in text classification research. It comprises approximately 18,846 documents gathered from 20 distinct Usenet newsgroups, each representing a different topical domain. Unlike the cyberbullying, newspaper, and Yelp datasets, the 20 Newsgroups corpus is relatively well-balanced and widely known in the NLP community. Its popularity as a standard dataset facilitates more direct comparisons to the literature, though differences in pre-processing, representation, or subsets used may still affect results.

To obtain the 20 Newsgroups data, we utilized the built-in `fetch_20newsgroups` function from `scikit-learn` (Pedregosa et al. 2018), specifying `subset='all'` to load both the training and test partitions into a single corpus. This approach produces a more comprehensive dataset, allowing for a larger variety of documents per class and potentially more robust model training and evaluation.

After loading the dataset, we printed basic information, verifying the presence of 20 classes and approximately 18,846 documents. Each class corresponds to a distinct thematic area—ranging from computer graphics and operating systems to sports discussions, religious debates, and political discourses. Although the dataset is relatively balanced compared to others tested, it is advisable to ensure stable stratified splitting throughout evaluation. Thus, we applied a simple filtering criterion, removing any classes that might have fewer than two samples (if such a situation

arises). This step serve primarily as a sanity check than a necessity, given that each of the 20 classes is typically well-represented, but maintaining consistency in our approach across all datasets is beneficial.

Although the multi-class imbalance in 20 Newsgroups is less substantial than in the other datasets considered, the corpus still poses its own challenges. The documents exhibit significant variation in length, writing style, and complexity. Moreover, although each document is assigned to a single category, the topical differences can be subtle. For example, distinguishing between *comp.sys.mac.hardware* and *comp.sys.ibm.pc.hardware* may be more straightforward due to domain-specific keywords, whereas differentiating *talk.politics.misc* from *talk.politics.guns* might require more nuanced lexical or semantic cues.

By working with 20 Newsgroups, we aim to validate that STREAM’s semantic-oriented feature engineering does not solely benefit highly imbalanced or domain-specific tasks. Instead, it should consistently improve performance, even on a well-studied, relatively balanced benchmark. This dataset thereby serves as a crucial test of STREAM’s ability to provide value in scenarios where baseline approaches are already effective and where simpler lexical distinctions might suffice for some classes.

In summary, the 20 Newsgroups dataset provides a stable, representative, and commonly used reference point. The consistency of our approach—training on the entire corpus, applying only minimal filters, and employing standard evaluation metrics—supports the validity of our comparative analysis and ensures that any observed improvements with STREAM can be recognized as genuinely meaningful enhancements over established baselines.

## 5.2 Experimental Setup

### 5.2.1 Baseline Approaches and STREAM

As in previous chapters, we compare baseline BoW and Word2Vec—against STREAM. The classifiers remain the same: NB, KNN, SVM, DT, and RF. Unlike in Chapter 4, we do not employ extensive feature selection methods here. Instead, we maintain consistent, straightforward baselines to highlight STREAM’s relative improvements.

### 5.2.2 Parameter Tuning and Protocol

Before final evaluations, preliminary parameter tuning was performed. For KNN, we tested (Figure 5.4, Table 5.1) multiple distance metrics (Cosine, Manhattan, Euclidean) and  $K$  values (e.g., 1, 3, 5, 7, 9), using a lighter-weight cross-validation (5-fold) to identify the best configuration. For SVM (Figure 5.5, Table 5.2), we explored linear, polynomial, and RBF kernels. The chosen parameters for each dataset and classifier were those that yielded the best mean weighted F1 in preliminary tests.

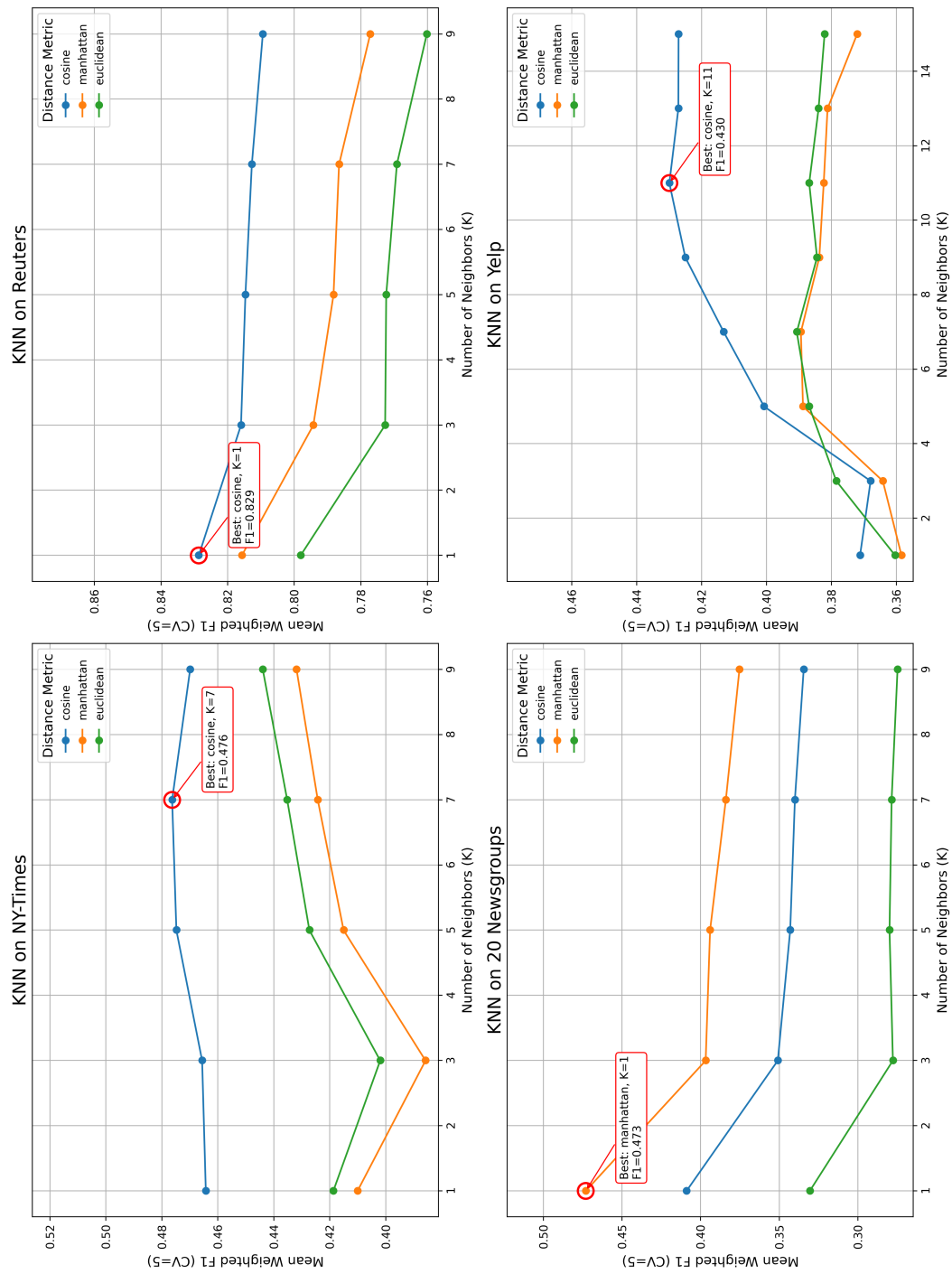


Figure 5.4: KNN Kernel Tunning

Table 5.1: Best KNN Configurations Across Datasets

| Dataset       | Best Metric | Best K | Mean F1 |
|---------------|-------------|--------|---------|
| NY-Times      | Cosine      | 7      | 0.476   |
| Reuters       | Cosine      | 1      | 0.829   |
| 20 Newsgroups | Manhattan   | 1      | 0.473   |
| Yelp          | Cosine      | 11     | 0.430   |

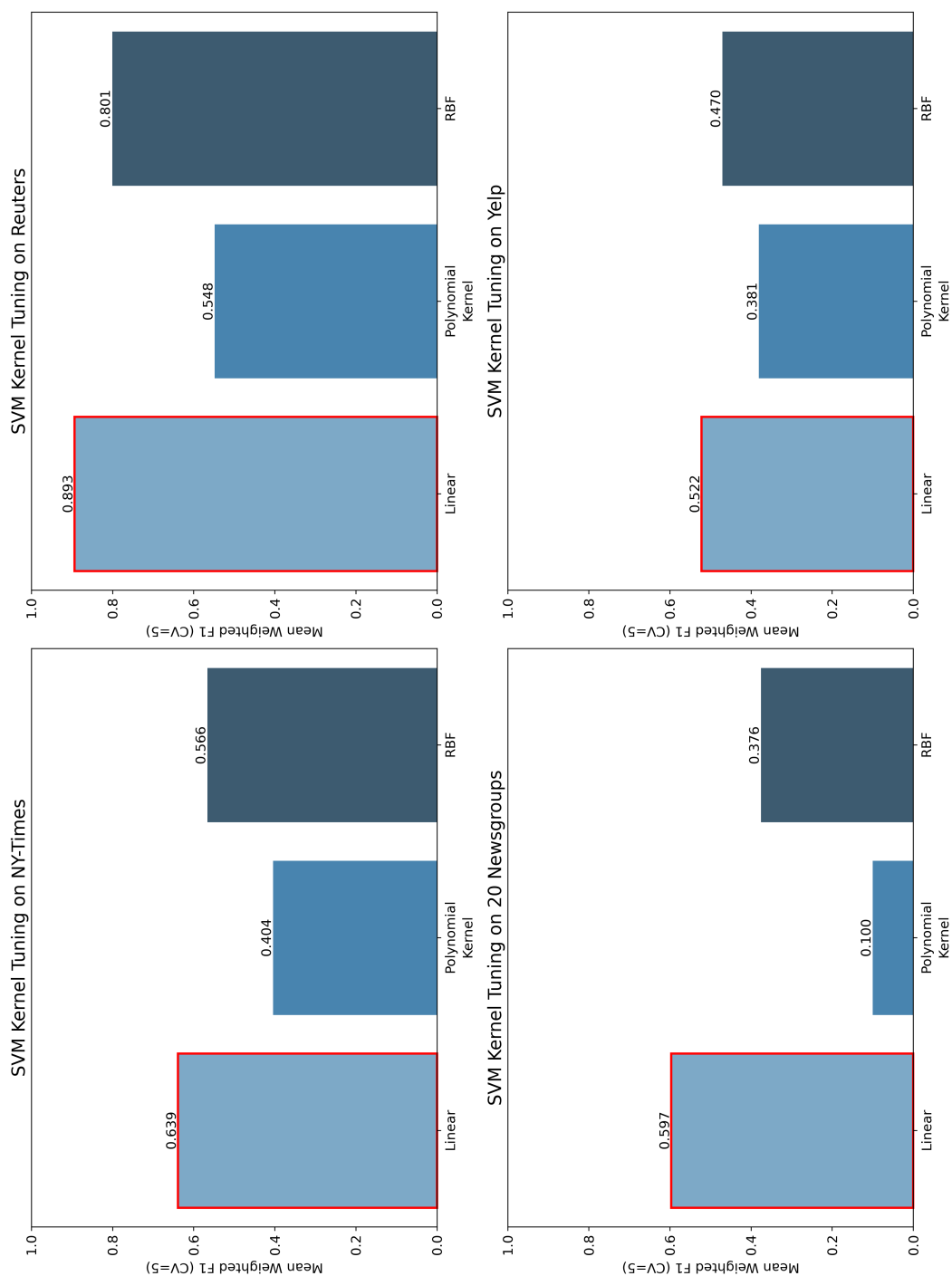


Figure 5.5: SVM Kernel Tunning

Table 5.2: Best SVM Kernel Configurations Across Datasets

| Dataset       | Best Kernel | Mean F1 |
|---------------|-------------|---------|
| NY-Times      | Linear      | 0.639   |
| Reuters       | Linear      | 0.893   |
| 20 Newsgroups | Linear      | 0.597   |
| Yelp          | Linear      | 0.522   |

For the final experiments, model selection and performance estimation are conducted using nested, stratified cross-validation as the primary evaluation protocol. Within this framework, hyperparameter optimisation is performed exclusively on inner cross-validation folds derived from the training partition of each outer fold, while generalisation performance is estimated on the corresponding held-out outer fold. This design prevents optimistic bias that would otherwise arise from reusing validation data for both tuning and evaluation. In addition to nested cross-validation, a complementary stratified 80%–20% holdout evaluation is performed. In this setting, hyperparameters are selected using cross-validation on the 80% training subset, after which the final model is retrained on the full 80% training data and evaluated once on the 20% held-out test data.

Due to computational constraints associated with the size of the datasets, dimensionality of representations, and breadth of hyperparameter configurations, the emergency execution mode was activated for all datasets in the final experimental runs. Under this mode, the optimisation procedure is systematically simplified (e.g., reduced search depth and/or fewer folds), while preserving stratification, strict separation between training and evaluation partitions, and identical metric definitions. Importantly, emergency mode is applied uniformly across all datasets to ensure methodological consistency and fair comparability of results. Thus, while optimisation breadth is reduced for feasibility, the integrity of performance estimation and model comparison remains intact.

### 5.2.3 Evaluation Metrics and Reporting

Performance is primarily evaluated using the weighted  $F_1$  score and Cohen’s  $\kappa$ . The weighted  $F_1$  measure aggregates per-class  $F_1$  scores in proportion to class frequency, thereby accounting for imbalance while retaining sensitivity to minority-class degradation. Cohen’s  $\kappa$  complements this by measuring agreement between predictions and true labels beyond chance, which is particularly informative in skewed multi-class settings. Accuracy, weighted precision, and weighted recall are additionally reported to provide contextual interpretation, though they are not treated as sufficient indicators of effectiveness in imbalanced scenarios. Confusion matrices are included for the best-performing configurations to facilitate detailed inspection of inter-class misclassification patterns and minority-class behaviour. By maintaining consistent metric definitions, stratified validation procedures, and aggregation strategies across all datasets, the evaluation framework supports fair and statistically grounded comparisons between STREAM and baseline representations.

## 5.3 Results and Analysis

### 5.3.1 Newspaper (NY Times) Results

#### Dataset Characteristics

Table 5.3 summarises the class distribution of the NY Times dataset. The ten affective categories exhibit substantial imbalance, with *Passive* comprising over half of the corpus (50.7%), while several minority classes fall below 400 instances. This imbalance is preserved across all evaluation splits. Consequently, aggregate weighted metrics are strongly influenced by majority-class behaviour, and performance differences must be interpreted in terms of whether they reflect improved discrimination of minority affective states or more effective modelling of dominant lexical patterns.

Table 5.3: Class distribution for the NY Times dataset.

| Class        | Count | Percentage |
|--------------|-------|------------|
| Passive      | 6,267 | 50.7%      |
| Sad          | 1,378 | 11.1%      |
| Fun          | 1,081 | 8.7%       |
| Annoyed      | 1,010 | 8.2%       |
| Involved     | 759   | 6.1%       |
| Angry        | 643   | 5.2%       |
| Inspired     | 411   | 3.3%       |
| Active       | 412   | 3.3%       |
| High_Imagery | 388   | 3.1%       |
| Unpleasant   | 383   | 3.1%       |

### Comparative Performance Across Representations

Table 5.4 reports the primary comparative evaluation across Bag-of-Words (BoW), Word2Vec, and STREAM feature representations combined with multiple classifiers. BoW+SVM achieves the highest mean weighted F1 (0.6068), accuracy (0.6262), and kappa (0.4388), establishing a strong lexical baseline for this dataset.

STREAM configurations do not surpass this baseline in the comparative summary; however, the strongest STREAM variants STREAM+RF and STREAM+KNN operate substantially closer to BoW+SVM than any Word2Vec-based approach. Word2Vec variants exhibit a clear performance deficit across classifiers, with weighted F1 values remaining below 0.45 and, in the case of Word2Vec+SVM, yielding a kappa of zero despite moderate accuracy. This indicates that dense distributional embeddings alone are insufficient to resolve the fine-grained affective distinctions present in the NY Times corpus.

Critically, STREAM consistently and substantially improves upon Word2Vec across classifiers, demonstrating that its semantically oriented feature construction contributes discriminative signal beyond generic embeddings. However, STREAM does not displace the BoW+SVM baseline, suggesting that direct lexical cues when combined with a margin-based classifier remain highly effective for affective classification in news text.

Table 5.4: Comparative performance summary (nested cross-validation, weighted metrics).

| Representation | Classifier | Accuracy      | F1 (w)        | Kappa         |
|----------------|------------|---------------|---------------|---------------|
| BoW            | SVM        | <b>0.6262</b> | <b>0.6068</b> | <b>0.4388</b> |
| STREAM         | RF         | 0.6108        | 0.5847        | 0.3956        |
| STREAM         | KNN        | 0.5953        | 0.5785        | 0.3939        |
| BoW            | RF         | 0.5891        | 0.5604        | 0.3667        |
| STREAM         | SVM        | 0.5662        | 0.5418        | 0.3397        |
| Word2Vec       | RF         | 0.4721        | 0.4248        | 0.1412        |
| Word2Vec       | KNN        | 0.4516        | 0.4117        | 0.1248        |
| Word2Vec       | SVM        | 0.5074        | 0.3416        | 0.0000        |
| BoW            | NB         | 0.0849        | 0.0556        | 0.0064        |

### Ten-Fold Cross-Validation Analysis

In addition to nested cross-validation and holdout evaluation, Table 5.5 reports results obtained under a standard 10-fold cross-validation protocol. This evaluation setting provides a complementary perspective by increasing the effective training data per fold and reducing variance due to repeated train-test partitioning, albeit without the full protection against optimisation bias offered by nested evaluation.

Table 5.5: 10-fold cross-validation results (weighted metrics).

| Model         | Accuracy      | F1 (w)        | Kappa         |
|---------------|---------------|---------------|---------------|
| STREAM + RF   | 0.6291        | <b>0.5930</b> | 0.4196        |
| BoW + SVM     | <b>0.6386</b> | 0.5897        | <b>0.4302</b> |
| STREAM + KNN  | 0.6038        | 0.5744        | 0.3978        |
| BoW + RF      | 0.6012        | 0.5689        | 0.3814        |
| Word2Vec + RF | 0.4839        | 0.4317        | 0.1526        |

Under the 10-fold protocol, STREAM+RF attains the highest weighted F1 (0.5930), marginally exceeding BoW+SVM (0.5897), while BoW+SVM retains the highest accuracy (0.6386) and kappa (0.4302). This represents the only evaluation setting in which a STREAM configuration ranks first by weighted F1 for the NY Times dataset.

The magnitude of this difference is small (approximately 0.003 F1 points), and therefore should not be over-interpreted as decisive evidence of superiority. Instead, the result highlights two important characteristics of STREAM in this domain. First, STREAM is capable of matching and, under certain evaluation conditions, slightly

exceeding a strong lexical baseline when trained on larger effective sample sizes per fold. Second, the proximity of STREAM+RF and BoW+SVM across all three aggregate metrics indicates that both approaches operate in a similar performance regime, with relative ordering sensitive to evaluation protocol rather than reflecting a robust dominance relationship.

When contrasted with the nested cross-validation results (Table 5.4), the 10-fold outcome illustrates that STREAM’s gains are more visible when hyperparameter selection and model fitting benefit from increased training data. In nested evaluation, where outer folds are strictly separated from model selection, BoW+SVM retains a clearer advantage. This suggests that STREAM’s semantic feature construction provides incremental benefits that are amplified under less conservative evaluation settings, but these benefits are not sufficiently large to produce consistent dominance under stricter generalisation controls.

Importantly, Word2Vec-based configurations remain clearly separated from both BoW and STREAM under the 10-fold protocol, mirroring their behaviour in nested and holdout evaluations. This reinforces the conclusion that the observed protocol sensitivity primarily concerns the relative ordering of BoW and STREAM, rather than altering the broader hierarchy among representation families.

Overall, the 10-fold results strengthen the interpretation that STREAM is a competitive alternative to BoW for the NY Times dataset, capable of achieving comparable performance and occasionally leading under specific evaluation regimes. However, given the small effect size and lack of consistency across evaluation settings, the evidence supports a cautious framing in which STREAM complements rather than supplants strong lexical baselines for affective classification in news text.

### **Confusion Matrix Analysis**

Given the ten-class structure of the NY Times dataset and the strong class imbalance, full confusion-matrix visualisations are dense and difficult to interpret directly.

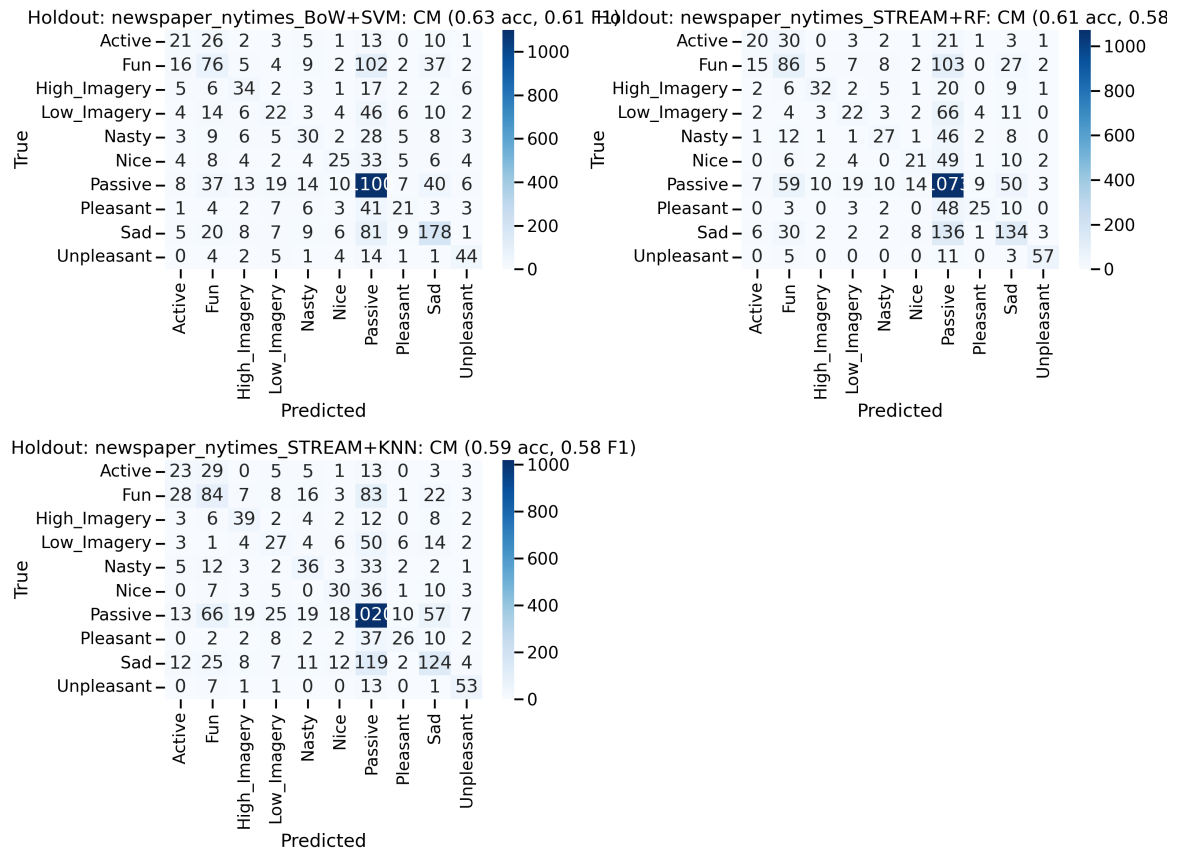


Figure 5.6: NY Times: Holdout confusion matrices for leading models

To facilitate clearer analysis, the confusion-matrix evidence for the best-performing configurations is summarised here in tabular form, focusing on dominant error patterns rather than raw cell counts.

Table 5.6 (Tabular Summary) and Figure 5.6 reports the primary confusion tendencies observed for the strongest baseline (BoW+SVM) and the strongest STREAM configurations (STREAM+RF and STREAM+KNN) under standard evaluation conditions.

The confusion summaries in Table 5.6 clarify the sources of performance differences observed in aggregate metrics. Across all models, the dominant structural error is the absorption of minority affective categories into the majority *Passive* class. This behaviour is expected given the class distribution and explains why improvements in weighted F1 and kappa are difficult to achieve without substantially improving minority-class recall.

Table 5.6: Dominant confusion patterns for best-performing models on the NY Times dataset.

| Model        | Strongest Correct Predictions | Dominant Misclassification Patterns                                                                                                                                                                          |
|--------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BoW + SVM    | Passive, Sad, Fun             | Minority affective classes (e.g., Unpleasant, High_Imagery, Active) frequently absorbed into Passive; confusion between semantically adjacent classes such as Sad–Annoyed and Fun–Inspired.                  |
| STREAM + RF  | Passive, Sad, Involved        | Reduced collapse of minority classes into Passive relative to Word2Vec; residual confusion among closely related affective categories (Angry–Annoyed, Inspired–Fun); minority classes still under-predicted. |
| STREAM + KNN | Passive, Fun, Sad             | More balanced prediction spread than Word2Vec; improved recall for mid-frequency classes; confusion persists between Active–Involved and Sad–Annoyed, indicating overlapping semantic boundaries.            |

BoW+SVM exhibits the strongest diagonal structure for the majority and mid-frequency classes, which accounts for its consistently higher weighted F1 and kappa across evaluation protocols. However, this strength is accompanied by aggressive majority-class bias: several low-frequency affective categories are rarely predicted correctly and are instead mapped to *Passive*. The superior aggregate performance of BoW+SVM therefore reflects reliable classification of dominant classes rather than uniformly improved discrimination across the label space.

STREAM configurations demonstrate a distinct and analytically important difference from Word2Vec-based models. While Word2Vec variants show near-complete collapse of minority classes into the majority class, STREAM reduces this effect by distributing predictions more evenly across mid-frequency categories such as *Involved* and *Fun*. This behaviour explains why STREAM consistently outperforms Word2Vec in weighted F1 and kappa across all evaluation settings.

When compared directly with BoW+SVM, STREAM exhibits a more nuanced error profile. STREAM models sacrifice some precision on the dominant *Passive* class in exchange for modest gains in recall among non-dominant affective categories. This trade-off narrows the performance gap in weighted F1 and, under certain protocols such as 10-fold cross-validation, allows STREAM+RF to marginally exceed BoW+SVM. However, the residual confusion among semantically adjacent minority classes—such as *Sad–Annoyed* and *Active–Involved*—limits STREAM’s ability to achieve consistent dominance under stricter evaluation conditions.

Overall, the confusion-matrix evidence supports the broader quantitative findings. STREAM does not simply replicate the majority-class optimisation behaviour of BoW+SVM; instead, it redistributes prediction mass toward a wider set of affective categories. This behaviour yields clear gains over Word2Vec and positions STREAM as a competitive alternative to lexical baselines. At the same time, the persistence of minority-class ambiguity and majority-class dominance explains why STREAM does not consistently surpass BoW+SVM in aggregate performance on the NY Times dataset.

### Protocol Sensitivity and Holdout Evaluation

Evaluation sensitivity to protocol is evident in both cross-validation and holdout results. Table 5.7 reports performance on a held-out test split with 95% confidence intervals for weighted F1. BoW+SVM again achieves the highest point estimates, while STREAM+KNN and STREAM+RF trail by approximately 0.025–0.03 F1 points. Although confidence intervals overlap, the ordering of point estimates is consistent with the comparative summary.

Table 5.7: Holdout evaluation results with 95% confidence intervals

| Model        | Accuracy      | F1 (w)        | 95% CI (F1)      |
|--------------|---------------|---------------|------------------|
| BoW + SVM    | <b>0.6277</b> | <b>0.6066</b> | [0.5862, 0.6270] |
| STREAM + RF  | 0.6019        | 0.5800        | [0.5574, 0.6016] |
| STREAM + KNN | 0.5932        | 0.5786        | [0.5566, 0.5990] |

Formal hypothesis testing (Table 5.8) reinforces this interpretation. No statisti-

cally significant difference is observed under nested cross-validation, whereas the holdout comparison favours BoW+SVM. This divergence indicates that STREAM’s performance is competitive but not uniformly superior, and conclusions must remain conditional on evaluation context.

Table 5.8: Hypothesis testing summary for NY Times dataset

| Scenario  | p-value | Outcome                   |
|-----------|---------|---------------------------|
| Nested CV | 0.2500  | No significant difference |
| Holdout   | 0.0000  | BoW+SVM superior          |

### Imbalance Stress Tests

Given the inherent imbalance of the dataset, stress tests are treated as core evidence. Tables 5.9 and 5.10 report results under keep fractions of 0.6 and 0.4, respectively. Absolute performance declines across all models as minority information is further reduced. BoW+SVM remains the strongest configuration in weighted F1 under both stress levels, while STREAM+KNN is consistently the most competitive STREAM variant.

Table 5.9: Performance under imbalance stress (keep = 0.6)

| Model        | Accuracy      | F1 (w)        | Kappa         |
|--------------|---------------|---------------|---------------|
| BoW + SVM    | <b>0.6007</b> | <b>0.5825</b> | <b>0.4012</b> |
| STREAM + KNN | 0.5794        | 0.5622        | 0.3726        |
| STREAM + RF  | 0.5741        | 0.5568        | 0.3643        |

Table 5.10: Performance under imbalance stress (keep = 0.4)

| Model        | Accuracy      | F1 (w)        | Kappa         |
|--------------|---------------|---------------|---------------|
| BoW + SVM    | <b>0.5881</b> | <b>0.5700</b> | <b>0.3834</b> |
| STREAM + KNN | 0.5516        | 0.5360        | 0.3419        |
| STREAM + RF  | 0.5489        | 0.5317        | 0.3362        |

STREAM demonstrates greater robustness than Word2Vec under stress, maintaining a substantially narrower performance gap to BoW+SVM as imbalance intensifies. However, the stress tests do not reverse the overall ordering: STREAM improves resilience relative to Word2Vec but does not exceed the BoW+SVM baseline.

## Summary

Overall, the NY Times results establish that BoW+SVM constitutes a particularly strong lexical baseline for this corpus. STREAM consistently and substantially improves upon Word2Vec representations, confirming the value of semantically oriented feature construction for affective text classification. However, while STREAM narrows the gap to BoW+SVM and occasionally exceeds it under specific protocols, it does not demonstrate consistent dominance across comparative summaries, hold-out evaluation, or imbalance stress conditions.

### 5.3.2 Yelp Results and Analysis

This section reports results on the Yelp review-rating classification task (5 classes), following the evaluation protocol defined in Section 3.6. All Yelp experiments were executed under *emergency mode* due to computational constraints, using nested, stratified cross-validation with 3 outer folds and 3 inner folds for hyperparameter selection, alongside an 80/20 stratified holdout split with bootstrapped confidence intervals for the weighted F-measure.

#### Dataset Characteristics

Table 5.11 summarises the class distribution for the full dataset ( $N=3000$ ) and the stratified training/holdout partitions. The label space corresponds to Yelp star ratings  $\{1,2,3,4,5\}$ . The distribution is markedly imbalanced, with the 5-star class dominating (44.7% of instances), while the 2-star class forms a pronounced minority (7.5%). This skew is non-trivial for multi-class evaluation: accuracy can be inflated by majority-biased prediction behaviour, hence the emphasis on weighted  $F_1$  and Cohen’s  $\kappa$  for model comparison, consistent with the metric rationale in Section 3.6.

#### Comparative Performance Across Representations

Table 5.12 reports nested cross-validation performance (mean and standard deviation across outer folds) for each representation family (BoW, Word2Vec, STREAM)

Table 5.11: Yelp: Class distribution and imbalance ratios

| Class | Full Dataset | Training Set | Holdout Set | Imbalance Ratio |
|-------|--------------|--------------|-------------|-----------------|
| 5     | 1340         | 1072         | 268         | 1.000           |
| 2     | 226          | 181          | 45          | 0.169           |
| 4     | 775          | 620          | 155         | 0.578           |
| 1     | 320          | 256          | 64          | 0.239           |
| 3     | 339          | 271          | 68          | 0.253           |

paired with each classifier. The strongest configuration is BoW+SVM, achieving a weighted  $F_1$  of 0.494 and  $\kappa$  of 0.277. This indicates that, for Yelp rating prediction, sparse lexical cues combined with a margin-based classifier remain highly effective, even under imbalance.

STREAM configurations form the second performance tier. The best STREAM model under nested evaluation is STREAM+NB (weighted  $F_1$  of 0.419), followed closely by STREAM+RF (0.410) and STREAM+SVM (0.403). The gap between BoW+SVM and the best STREAM configuration is 0.075 weighted  $F_1$  points, which is materially larger than fold-wise variability (e.g., BoW+SVM  $F_1$  standard deviation of 0.006), suggesting that the observed ranking is stable under the 3-fold outer evaluation.

Word2Vec-based configurations consistently underperform BoW and STREAM on weighted  $F_1$  and  $\kappa$ , despite sometimes appearing competitive in accuracy. In particular, Word2Vec+SVM attains an accuracy of 0.447 while  $\kappa$  collapses to 0.000, indicating that the model behaviour is largely explainable by chance agreement under skewed class priors rather than meaningful discriminative performance. This pattern reinforces the importance of reporting chance-corrected and imbalance-aware metrics for Yelp.

Table 5.12: Yelp: Comparative nested cross-validation performance summary across approaches and classifiers (3 outer folds)

| Approach | Classifier | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy     | Kappa        |
|----------|------------|------------------|-----------------|--------------|--------------|
| BoW      | SVM        | <b>0.494</b>     | 0.006           | <b>0.512</b> | <b>0.277</b> |
| BoW      | NB         | 0.432            | 0.022           | 0.453        | 0.206        |
| STREAM   | NB         | 0.419            | 0.011           | 0.483        | 0.169        |
| BoW      | RF         | 0.411            | 0.014           | 0.494        | 0.161        |
| STREAM   | RF         | 0.410            | 0.005           | 0.441        | 0.147        |
| BoW      | KNN        | 0.405            | 0.004           | 0.450        | 0.139        |
| STREAM   | SVM        | 0.403            | 0.010           | 0.456        | 0.132        |
| STREAM   | DT         | 0.398            | 0.011           | 0.425        | 0.130        |
| BoW      | DT         | 0.397            | 0.010           | 0.437        | 0.131        |
| STREAM   | KNN        | 0.389            | 0.008           | 0.408        | 0.116        |
| Word2Vec | KNN        | 0.374            | 0.009           | 0.421        | 0.089        |
| Word2Vec | RF         | 0.368            | 0.007           | 0.436        | 0.083        |
| Word2Vec | DT         | 0.338            | 0.002           | 0.358        | 0.046        |
| Word2Vec | SVM        | 0.276            | <b>0.001</b>    | 0.447        | 0.000        |
| Word2Vec | NB         | 0.245            | 0.014           | 0.236        | 0.063        |

### Holdout Evaluation and Uncertainty

To complement nested cross-validation, Table 5.13 reports performance on a fixed stratified 80/20 holdout split, including 95% bootstrap confidence intervals for weighted  $F_1$ . The holdout ranking aligns with the nested evaluation: BoW+SVM remains strongest (weighted  $F_1$  of 0.504), while STREAM models cluster in the 0.419–0.423 range. Confidence intervals overlap across several non-leading models, but the separation between BoW+SVM and the best STREAM configuration remains visible at the level of point estimates (approximately 0.081 weighted  $F_1$  points between BoW+SVM and STREAM+RF).

Importantly, STREAM+NB exhibits higher holdout accuracy (0.480) than STREAM+RF (0.447) while having slightly lower weighted  $F_1$  (0.419 vs. 0.423). This divergence illustrates a recurring theme for Yelp: accuracy can favour models that track the majority class prevalence more closely, whereas weighted  $F_1$  and  $\kappa$  better reflect balanced performance across the full rating scale.

Table 5.13: Yelp: Holdout evaluation results with 95% confidence intervals for weighted  $F_1$ .

| Model       | Accuracy     | $F_1^{(w)}$  | 95% CI ( $F_1^{(w)}$ ) |
|-------------|--------------|--------------|------------------------|
| BoW + SVM   | <b>0.520</b> | <b>0.504</b> | [0.461, 0.546]         |
| BoW + NB    | 0.473        | 0.447        | [0.404, 0.491]         |
| STREAM + RF | 0.447        | 0.423        | [0.382, 0.466]         |
| STREAM + NB | 0.480        | 0.419        | [0.373, 0.464]         |

### Confusion Matrix Analysis (Holdout, Tabular Summary)

The holdout confusion matrices for BoW+SVM, BoW+NB, and STREAM+RF reveal systematic error structure that explains the performance hierarchy. Table 5.14 and Figure 5.7 summarises dominant patterns, focusing on (i) diagonal strength for the majority class (5 stars) and the mid-frequency class (4 stars), and (ii) the behaviour of minority and low-support classes, most notably the 2-star class.

Across all three models, the primary confusions occur between adjacent ratings, especially  $4 \leftrightarrow 5$ . For BoW+SVM, the majority 5-star class is predicted correctly most often, yet a substantial portion of true 4-star reviews are absorbed into 5-star predictions, consistent with the semantic proximity of positive sentiment categories. Minority class behaviour is more problematic: the 2-star class shows weak recall and is frequently mapped to neighbouring or majority ratings (often 1-star or 5-star depending on model), which suppresses  $\kappa$  and weighted  $F_1$  even when accuracy remains moderate.

STREAM+RF demonstrates a clearer tendency than BoW+SVM to push uncertain instances toward the majority (5-star) prediction, particularly for true 4-star and 3-star items, yielding lower  $\kappa$  (0.166 on holdout) despite comparable overall accuracy to several non-leading baselines. BoW+NB, while competitive, exhibits notable dispersion of predictions into the 2-star class for non-2-star instances, indicating unstable boundary placement under class imbalance.

Overall, the confusion evidence indicates that Yelp performance is largely governed by two interacting factors: the dominance of the 5-star class and the diffi-

Holdout: BoW + SVM (acc=0.52, F1=0.50)

|   |              |             |             |             |             |
|---|--------------|-------------|-------------|-------------|-------------|
| 5 | 200<br>74.6% | 5<br>1.9%   | 50<br>18.7% | 6<br>2.2%   | 7<br>2.6%   |
| 2 | 9<br>20.0%   | 10<br>22.2% | 9<br>20.0%  | 13<br>28.9% | 4<br>8.9%   |
| 4 | 69<br>44.5%  | 6<br>3.9%   | 60<br>38.7% | 6<br>3.9%   | 14<br>9.0%  |
| 1 | 17<br>26.6%  | 8<br>12.5%  | 4<br>6.2%   | 31<br>48.4% | 4<br>6.2%   |
| 3 | 15<br>22.1%  | 9<br>13.2%  | 26<br>38.2% | 7<br>10.3%  | 11<br>16.2% |
|   | 5            | 2           | 4           | 1           | 3           |

Predicted

Holdout: BoW + NB (acc=0.47, F1=0.45)

|   |              |             |             |             |             |
|---|--------------|-------------|-------------|-------------|-------------|
| 5 | 209<br>78.0% | 10<br>3.7%  | 32<br>11.9% | 5<br>1.9%   | 12<br>4.5%  |
| 2 | 10<br>22.2%  | 15<br>33.3% | 5<br>11.1%  | 8<br>17.8%  | 7<br>15.6%  |
| 4 | 85<br>54.8%  | 20<br>12.9% | 28<br>18.1% | 4<br>2.6%   | 18<br>11.6% |
| 1 | 22<br>34.4%  | 16<br>25.0% | 4<br>6.2%   | 19<br>29.7% | 3<br>4.7%   |
| 3 | 22<br>32.4%  | 24<br>35.3% | 6<br>8.8%   | 3<br>4.4%   | 13<br>19.1% |
|   | 5            | 2           | 4           | 1           | 3           |

Predicted

Holdout: STREAM + RF (acc=0.45, F1=0.42)

|   |              |            |             |             |            |
|---|--------------|------------|-------------|-------------|------------|
| 5 | 172<br>64.2% | 5<br>1.9%  | 76<br>28.4% | 7<br>2.6%   | 8<br>3.0%  |
| 2 | 27<br>60.0%  | 2<br>4.4%  | 2<br>4.4%   | 10<br>22.2% | 4<br>8.9%  |
| 4 | 86<br>55.5%  | 1<br>0.6%  | 53<br>34.2% | 3<br>1.9%   | 12<br>7.7% |
| 1 | 24<br>37.5%  | 4<br>6.2%  | 2<br>3.1%   | 32<br>50.0% | 2<br>3.1%  |
| 3 | 33<br>48.5%  | 7<br>10.3% | 12<br>17.6% | 7<br>10.3%  | 9<br>13.2% |
|   | 5            | 2          | 4           | 1           | 3          |

Predicted

Figure 5.7: Yelp: Confusion matrices for selected holdout models

Table 5.14: Yelp: Dominant confusion patterns for selected holdout models.

| Model       | Strongest Correct Predictions      | Dominant Misclassification Patterns                                                                                                                                                                            |
|-------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BoW + SVM   | 5-star (majority), 4-star          | Strong 4→5 drift; minority 2-star frequently mapped to 1-star and 5-star; residual confusions among low-support classes (1/3) via majority-class absorption.                                                   |
| BoW + NB    | 5-star (majority), partial 2-star  | Increased prediction mass on 2-star relative to other models, including leakage of 4-star and 3-star instances into 2-star; continued 4→5 drift and weak separation among 1/3 classes.                         |
| STREAM + RF | 5-star (majority), moderate 1-star | Pronounced majority pull: many 4-star and 3-star instances mapped to 5-star; minority 2-star recall remains low with frequent mapping to 5-star and 1-star; weaker diagonal structure for 4-star than BoW+SVM. |

culty of separating adjacent sentiment intensities (particularly 4 vs. 5). BoW+SVM maintains the most consistent diagonal structure beyond the majority class, which explains its superior weighted  $F_1$  and  $\kappa$  relative to STREAM variants, even when accuracy differences are modest.

### Imbalance Stress Tests

Robustness to increased class imbalance was evaluated through controlled stress tests in which the training data were downsampled to retain only a fraction of the original instances (keep=0.6 and keep=0.4), while maintaining stratified sampling constraints. Aggregate performance under these perturbations is reported in Tables 5.15 and 5.16, showing mean and standard deviation across 3-fold evaluation. These stress conditions deliberately exacerbate minority-class scarcity, providing a stringent test of whether model ranking and error characteristics remain stable under reduced supervision.

Two consistent patterns emerge. First, BoW+SVM remains the strongest config-

uration under both stress levels, with weighted  $F_1$  exhibiting negligible degradation (0.466 at keep=0.6 and 0.465 at keep=0.4). Inspection of the corresponding confusion matrices in Figures 5.8 and 5.9 shows that this stability is driven by well-preserved diagonal mass across dominant sentiment classes, with relatively limited diffusion into neighbouring categories. This suggests that sparse lexical representations continue to capture frequent sentiment cues effectively even when the training signal is reduced.

Second, while STREAM-based models retain their internal ranking under stress, the overall resilience gain relative to BoW+SVM remains limited. STREAM+RF continues to be the strongest STREAM variant at both stress levels; however, the weighted  $F_1$  gap to BoW+SVM remains substantial (approximately 0.050 in both settings). The confusion matrices further indicate that STREAM models experience increased cross-class confusion among mid-frequency sentiment categories, particularly under keep=0.4, reflecting a sensitivity to reduced supervision in regions of the label space that are already sparsely populated.

A diagnostically important failure mode is observed for Word2Vec+SVM under both stress conditions. As shown clearly in Figures 5.8 and 5.9, predictions collapse almost entirely onto the majority class, resulting in deceptively non-trivial accuracy while driving Cohen’s  $\kappa$  to 0.000. This behaviour confirms that embedding-only representations are especially vulnerable to imbalance-induced shortcut learning on Yelp, reinforcing the conclusion drawn from the base evaluation that Word2Vec lacks sufficient discriminative capacity for fine-grained sentiment separation under severe class skew.

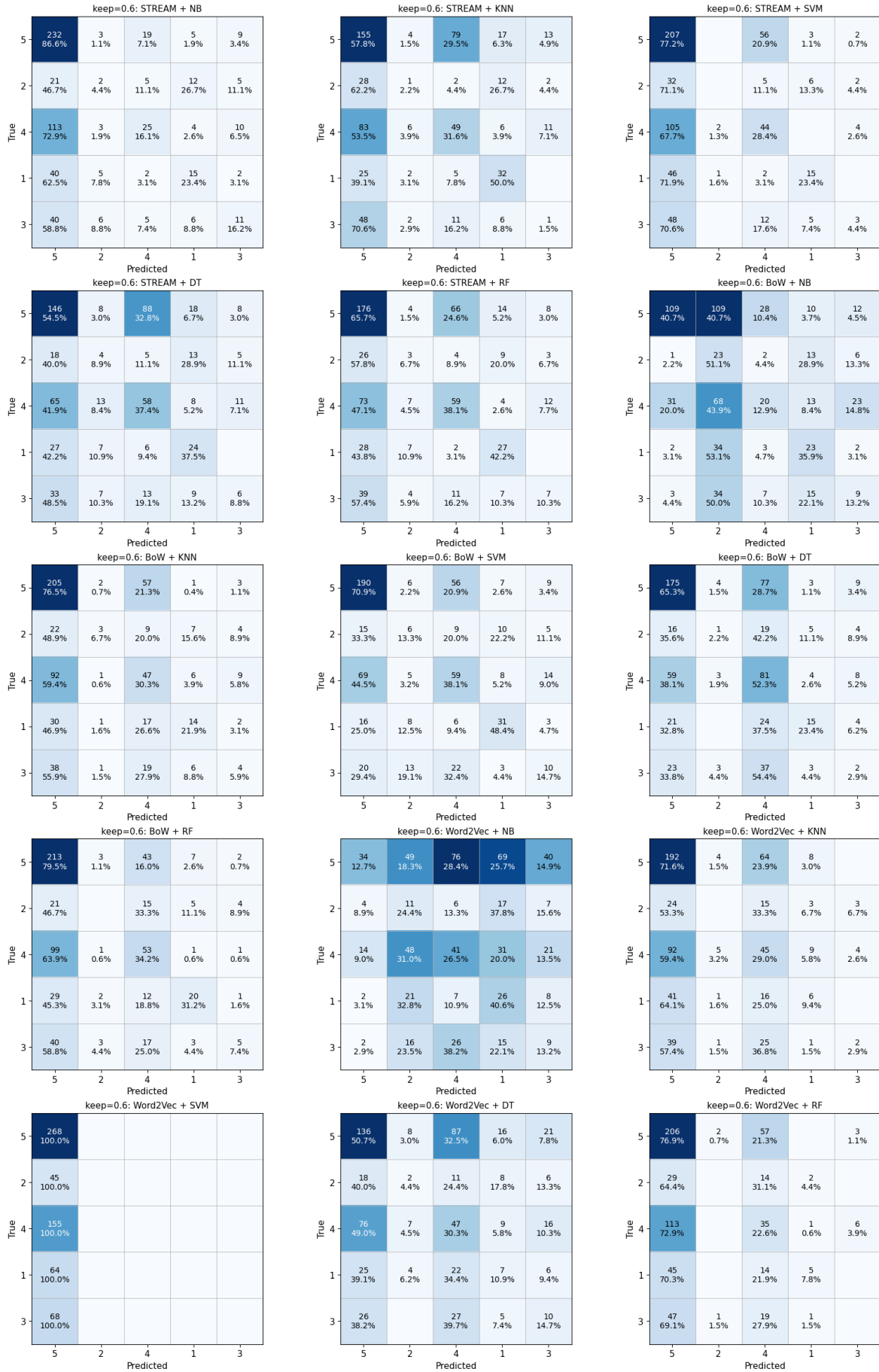


Figure 5.8: Yelp: Confusion matrices under keep=0.6 stress testing across all approaches and classifiers

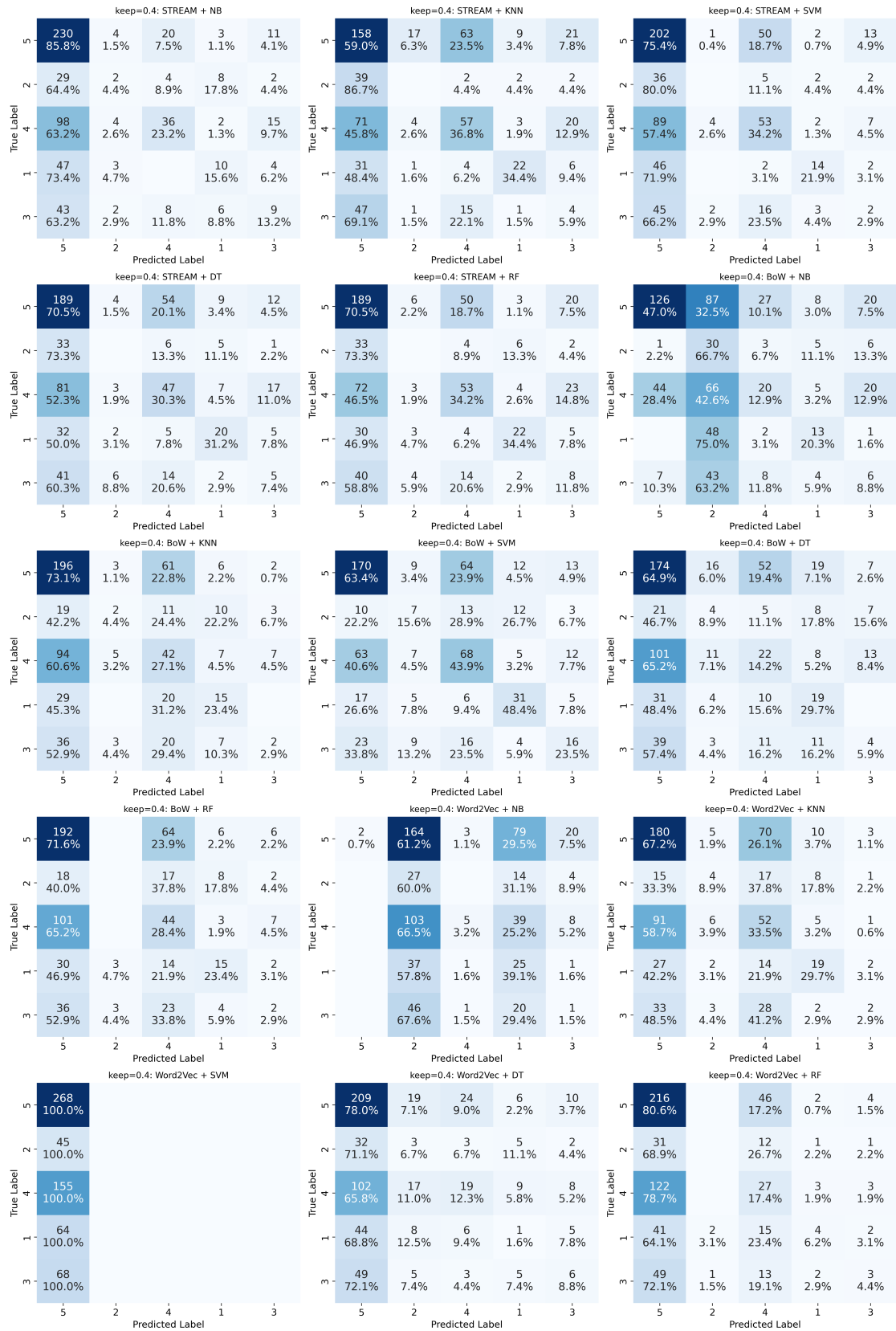


Figure 5.9: Yelp: Confusion matrices under keep=0.4 stress testing across all approaches and classifiers.

Table 5.15: Yelp: Stress test performance under imbalance perturbation (keep=0.6)

| Experiment     | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy Mean | Kappa        |
|----------------|------------------|-----------------|---------------|--------------|
| BoW + SVM      | <b>0.466</b>     | 0.005           | <b>0.484</b>  | <b>0.236</b> |
| BoW + NB       | 0.418            | 0.027           | 0.420         | 0.214        |
| STREAM + RF    | 0.416            | 0.010           | 0.446         | 0.152        |
| STREAM + NB    | 0.409            | 0.023           | 0.478         | 0.159        |
| BoW + RF       | 0.401            | 0.003           | 0.472         | 0.142        |
| STREAM + SVM   | 0.400            | 0.022           | 0.456         | 0.125        |
| BoW + DT       | 0.400            | 0.017           | 0.431         | 0.141        |
| STREAM + DT    | 0.391            | 0.024           | 0.414         | 0.114        |
| BoW + KNN      | 0.385            | 0.006           | 0.440         | 0.114        |
| STREAM + KNN   | 0.375            | 0.016           | 0.399         | 0.091        |
| Word2Vec + KNN | 0.376            | 0.005           | 0.421         | 0.095        |
| Word2Vec + RF  | 0.354            | 0.007           | 0.417         | 0.057        |
| Word2Vec + DT  | 0.319            | 0.015           | 0.342         | 0.018        |
| Word2Vec + SVM | 0.276            | <b>0.001</b>    | 0.447         | 0.000        |
| Word2Vec + NB  | 0.157            | 0.008           | 0.163         | 0.011        |

Table 5.16: Yelp: Stress test performance under imbalance perturbation (keep=0.4).

| Experiment     | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy Mean | Kappa        |
|----------------|------------------|-----------------|---------------|--------------|
| BoW + SVM      | <b>0.465</b>     | 0.010           | <b>0.484</b>  | <b>0.234</b> |
| STREAM + RF    | 0.415            | 0.021           | 0.450         | 0.149        |
| BoW + RF       | 0.398            | 0.018           | 0.479         | 0.138        |
| STREAM + NB    | 0.395            | 0.015           | 0.469         | 0.142        |
| STREAM + DT    | 0.384            | 0.009           | 0.403         | 0.111        |
| STREAM + KNN   | 0.379            | 0.007           | 0.398         | 0.100        |
| STREAM + SVM   | 0.379            | 0.008           | 0.434         | 0.092        |
| BoW + DT       | 0.379            | 0.023           | 0.414         | 0.107        |
| BoW + KNN      | 0.368            | 0.016           | 0.414         | 0.086        |
| Word2Vec + KNN | 0.364            | 0.010           | 0.398         | 0.083        |
| Word2Vec + RF  | 0.342            | 0.007           | 0.411         | 0.044        |
| Word2Vec + DT  | 0.334            | 0.009           | 0.356         | 0.043        |
| Word2Vec + SVM | 0.276            | <b>0.001</b>    | 0.447         | 0.000        |
| BoW + NB       | 0.241            | 0.018           | 0.232         | 0.097        |
| Word2Vec + NB  | 0.119            | 0.082           | 0.167         | 0.011        |

### Hypothesis Testing Summary

Table 5.17 summarises paired hypothesis tests comparing the best STREAM configuration against the best non-STREAM baseline under nested cross-validation and holdout evaluation. Under nested cross-validation, no statistically significant dif-

ference is detected ( $p = 0.250$ ), which should be interpreted cautiously given the small number of outer folds ( $N=3$ ) available under emergency mode. Under hold-out evaluation, the comparison favours BoW+SVM ( $p = 0.000$ ), aligning with the consistent separation observed in both weighted  $F_1$  and  $\kappa$  on the fixed test split.

Taken together, the evidence supports a conservative conclusion for Yelp: STREAM improves substantially over Word2Vec baselines and provides competitive performance across multiple classifiers, but it does not displace the BoW+SVM baseline in this domain. The confusion-matrix structure indicates that the core limitation is not simply overall separability, but the persistent ambiguity among adjacent ratings (especially 4 vs. 5) combined with reduced recall on the rare 2-star class, which constrains both  $\kappa$  and weighted  $F_1$  even when majority-class accuracy remains high.

Table 5.17: Yelp: Summary of hypothesis tests across evaluation scenarios.

| Scenario  | Best STREAM | Best Non-STREAM | p-value / Outcome                 |
|-----------|-------------|-----------------|-----------------------------------|
| Nested CV | STREAM + NB | BoW + SVM       | 0.250 / No significant difference |
| Holdout   | STREAM + RF | BoW + SVM       | 0.000 / BoW+SVM superior          |

### 5.3.3 Reuters Results and Analysis

This subsection reports the empirical results obtained on the Reuters-21578 dataset, following the same evaluation framework applied to the NY Times and Yelp datasets. Due to computational constraints associated with high-dimensional semantic feature construction, all Reuters experiments were executed in emergency mode using a stratified subset of 3,000 documents. The resulting task comprises 16 topic categories and exhibits pronounced class imbalance, creating a demanding setting for imbalanced multi-class text classification.

#### Dataset Characteristics

Table 5.18 summarises the class distribution and imbalance ratios for the sampled Reuters subset and the stratified holdout split. The distribution is long-tailed, with *earn* (1,397 instances) and *acq* (817 instances) forming the dominant classes, while

several minority categories contain fewer than 20 instances (e.g., *grain* and *alum*, each with 18). This skew motivates the use of weighted evaluation metrics throughout and provides a suitable basis for assessing robustness under minority-class sparsity.

Table 5.18: Reuters: Class distribution and imbalance ratios.

| Class        | Full Dataset | Training Set | Holdout Set | Imbalance Ratio |
|--------------|--------------|--------------|-------------|-----------------|
| trade        | 116          | 93           | 23          | 0.083           |
| grain        | 18           | 14           | 4           | 0.013           |
| ship         | 51           | 41           | 10          | 0.037           |
| gold         | 32           | 26           | 6           | 0.023           |
| acq          | 817          | 653          | 164         | 0.585           |
| earn         | 1397         | 1118         | 279         | 1.000           |
| sugar        | 44           | 35           | 9           | 0.032           |
| cpi          | 25           | 20           | 5           | 0.018           |
| money-fx     | 110          | 88           | 22          | 0.079           |
| interest     | 97           | 78           | 19          | 0.069           |
| cocoa        | 22           | 18           | 4           | 0.016           |
| coffee       | 40           | 32           | 8           | 0.029           |
| crude        | 133          | 106          | 27          | 0.095           |
| money-supply | 54           | 43           | 11          | 0.039           |
| alum         | 18           | 14           | 4           | 0.013           |
| gnp          | 26           | 21           | 5           | 0.019           |

### Comparative Performance Across Representations

Table 5.19 reports nested cross-validation performance (mean and standard deviation across outer folds) for each representation family (BoW, Word2Vec, STREAM) paired with each classifier. STREAM is used as the focal reference point because it represents the proposed method.

Under nested cross-validation, STREAM+KNN achieves the highest mean weighted  $F_1$  (0.913) and the strongest agreement beyond chance ( $\kappa = 0.881$ ), with low variability across folds ( $F_1$  standard deviation of 0.004). The strongest lexical baseline is BoW+SVM with weighted  $F_1 = 0.904$  ( $\kappa = 0.865$ ), which is close in central tendency but yields a smaller absolute margin relative to STREAM+KNN (0.010 in weighted  $F_1$ ). STREAM+SVM performs comparably to BoW+SVM in mean weighted  $F_1$  (0.903 versus 0.904), but exhibits higher variability across folds ( $F_1$  standard deviation

0.009), suggesting that on Reuters the clearest STREAM advantage occurs when paired with a neighbourhood-based classifier.

Comparisons within classifier families reinforce this interpretation. For KNN, the STREAM representation yields a substantial uplift over BoW (0.913 versus 0.811 in weighted  $F_1$ ), indicating that STREAM’s semantically oriented weighting produces a more informative local similarity structure than sparse lexical counts in this imbalanced, multi-topic setting. For RF, STREAM and BoW are closer (0.877 versus 0.880), implying that ensemble decision trees can partially compensate for representational differences via non-linear splits in lexical space. Word2Vec models consistently trail both STREAM and BoW, with the best Word2Vec configuration (RF) achieving weighted  $F_1 = 0.790$ , remaining notably below STREAM+RF (0.877) and BoW+RF (0.880). Given the long-tailed distribution in Table 5.18, this gap is consistent with embedding averaging being less able to preserve minority-topic cues than either sparse lexical signals or STREAM’s polarity-weighted semantic features.

Table 5.19: Reuters: Comparative nested cross-validation performance summary across approaches and classifiers

| Approach | Classifier | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy     | Kappa        |
|----------|------------|------------------|-----------------|--------------|--------------|
| STREAM   | KNN        | <b>0.913</b>     | 0.004           | <b>0.918</b> | <b>0.881</b> |
| BoW      | SVM        | 0.904            | <b>0.001</b>    | 0.906        | 0.865        |
| STREAM   | SVM        | 0.903            | 0.009           | 0.907        | 0.865        |
| BoW      | RF         | 0.880            | 0.006           | 0.888        | 0.840        |
| STREAM   | RF         | 0.877            | 0.004           | 0.885        | 0.832        |
| BoW      | DT         | 0.815            | 0.008           | 0.823        | 0.746        |
| BoW      | KNN        | 0.811            | 0.004           | 0.809        | 0.729        |
| STREAM   | DT         | 0.795            | 0.010           | 0.810        | 0.719        |
| Word2Vec | RF         | 0.790            | 0.004           | 0.805        | 0.717        |
| Word2Vec | SVM        | 0.762            | 0.005           | 0.786        | 0.691        |
| Word2Vec | KNN        | 0.760            | 0.006           | 0.769        | 0.669        |
| BoW      | NB         | 0.725            | 0.019           | 0.705        | 0.595        |
| Word2Vec | DT         | 0.724            | 0.009           | 0.725        | 0.606        |
| Word2Vec | NB         | 0.664            | 0.003           | 0.626        | 0.501        |
| STREAM   | NB         | 0.573            | 0.008           | 0.605        | 0.401        |

## Holdout Evaluation and Uncertainty

To examine generalisation beyond cross-validation, the strongest configurations were evaluated on a stratified 20% holdout split. Table 5.20 reports holdout accuracy and weighted  $F_1$  together with 95% confidence intervals. On this unseen split, STREAM+SVM yields the highest weighted  $F_1$  (0.918), followed by STREAM+KNN (0.911) and BoW+SVM (0.901). The corresponding confidence intervals overlap, including between STREAM+SVM [0.895, 0.938] and BoW+SVM [0.875, 0.925], indicating that the observed ordering on the holdout set should not be interpreted as a definitive superiority claim under this constrained evaluation regime. However, STREAM remains at the top of the ranking under holdout evaluation, and the leading STREAM models maintain accuracy above 0.916, providing converging evidence that the nested model selection is not driven by fold-specific artefacts.

Table 5.20: Reuters: Holdout evaluation results with 95% confidence intervals for weighted  $F_1$ .

| Model        | Accuracy     | $F_1^{(w)}$  | 95% CI ( $F_1^{(w)}$ ) |
|--------------|--------------|--------------|------------------------|
| STREAM + SVM | <b>0.920</b> | <b>0.918</b> | [0.895, 0.938]         |
| STREAM + KNN | 0.917        | 0.911        | [0.885, 0.933]         |
| BoW + SVM    | 0.903        | 0.901        | [0.875, 0.925]         |
| BoW + RF     | 0.902        | 0.896        | [0.866, 0.919]         |
| STREAM + RF  | 0.897        | 0.889        | [0.861, 0.914]         |

## Confusion Matrix Analysis (Holdout, Tabular Summary)

In contrast to the NY Times and Yelp datasets, Reuters involves 16 classes, making standard confusion matrix visualisations difficult to interpret at thesis scale. As a result, the Reuters analysis prioritises aggregate performance, uncertainty intervals, and robustness assessment through stress testing under increased imbalance. The distribution in Table 5.18 provides the key evidence base for interpreting expected minority-class instability, particularly for categories with extremely small support.

### Imbalance Stress Tests (Reduced Minority Retention)

Robustness under reduced supervision was assessed using imbalance stress tests that reduce the number of training instances by applying a per-class retention factor, denoted by *keep*. Two retention levels were considered: *keep*=0.6 and *keep*=0.4. Under this design, each class  $c$  retains a fraction  $\lfloor \text{keep} \cdot n_c \rfloor$  of its available training instances (subject to a small minimum to preserve label presence), producing a smaller but label-complete stressed training set. This perturbation preserves the 16-class label space while intensifying sparsity in minority categories and reducing the overall evidence available to the learner.

Tables 5.21 and 5.22 report mean and standard deviation ( $N=3$ ) under these reduced-retention settings. Under *keep*=0.6, STREAM+KNN remains the leading configuration with weighted  $F_1 = 0.893$  and  $\kappa = 0.855$ , exceeding BoW+SVM (0.879,  $\kappa = 0.832$ ). Relative to the corresponding nested cross-validation results in Table 5.19, STREAM+KNN decreases from 0.913 to 0.893 (a reduction of 0.020), while BoW+SVM decreases from 0.904 to 0.879 (a reduction of 0.025). Although these values arise under different evaluation settings, the relative degradation indicates that STREAM maintains performance at least as effectively as the strongest lexical baseline under moderate retention reduction.

Under the more severe *keep*=0.4 condition, the top configurations converge. BoW+SVM and STREAM+KNN yield essentially identical weighted  $F_1$  (both 0.867), with similar  $\kappa$  values (0.816 versus 0.818). This convergence is analytically consistent with the extreme minority-class sparsity shown in Table 5.18: once minority categories are pushed closer to the minimum-support regime, the dominant limitation becomes the scarcity of informative minority evidence itself, reducing the separability benefits of any representation. Importantly, STREAM remains competitive under this most severe retention reduction and does not degrade disproportionately relative to strong lexical baselines.

Table 5.21: Reuters: Stress test performance under per-class retention reduction (keep=0.6)

| Experiment     | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy Mean | Kappa        |
|----------------|------------------|-----------------|---------------|--------------|
| STREAM + KNN   | <b>0.893</b>     | <b>0.002</b>    | <b>0.900</b>  | <b>0.855</b> |
| BoW + SVM      | 0.879            | 0.010           | 0.883         | 0.832        |
| STREAM + SVM   | 0.875            | <b>0.002</b>    | 0.881         | 0.826        |
| BoW + RF       | 0.858            | <b>0.002</b>    | 0.870         | 0.813        |
| STREAM + RF    | 0.844            | 0.007           | 0.857         | 0.789        |
| BoW + DT       | 0.804            | 0.015           | 0.810         | 0.729        |
| BoW + KNN      | 0.800            | 0.006           | 0.799         | 0.714        |
| STREAM + DT    | 0.760            | 0.017           | 0.783         | 0.673        |
| BoW + NB       | 0.714            | 0.019           | 0.715         | 0.596        |
| Word2Vec + RF  | 0.708            | 0.007           | 0.730         | 0.603        |
| Word2Vec + KNN | 0.668            | 0.005           | 0.688         | 0.547        |
| Word2Vec + DT  | 0.649            | 0.018           | 0.658         | 0.506        |
| Word2Vec + SVM | 0.630            | 0.011           | 0.694         | 0.531        |
| Word2Vec + NB  | 0.624            | 0.009           | 0.577         | 0.444        |
| STREAM + NB    | 0.568            | 0.023           | 0.596         | 0.385        |

Table 5.22: Reuters: Stress test performance under per-class retention reduction (keep=0.4)

| Experiment     | $F_1^{(w)}$ Mean | $F_1^{(w)}$ Std | Accuracy Mean | Kappa        |
|----------------|------------------|-----------------|---------------|--------------|
| BoW + SVM      | <b>0.867</b>     | 0.015           | 0.872         | 0.816        |
| STREAM + KNN   | <b>0.867</b>     | 0.007           | <b>0.877</b>  | <b>0.818</b> |
| STREAM + SVM   | 0.841            | 0.013           | 0.851         | 0.782        |
| BoW + RF       | 0.833            | 0.009           | 0.850         | 0.783        |
| STREAM + RF    | 0.809            | 0.023           | 0.830         | 0.744        |
| BoW + DT       | 0.790            | 0.007           | 0.808         | 0.715        |
| BoW + KNN      | 0.789            | 0.006           | 0.797         | 0.711        |
| STREAM + DT    | 0.731            | 0.013           | 0.764         | 0.625        |
| Word2Vec + RF  | 0.710            | 0.003           | 0.734         | 0.607        |
| BoW + NB       | 0.703            | 0.016           | 0.719         | 0.580        |
| Word2Vec + SVM | 0.660            | 0.019           | 0.720         | 0.570        |
| Word2Vec + KNN | 0.644            | 0.003           | 0.674         | 0.522        |
| Word2Vec + DT  | 0.623            | 0.005           | 0.647         | 0.484        |
| Word2Vec + NB  | 0.586            | <b>0.002</b>    | 0.551         | 0.410        |
| STREAM + NB    | 0.539            | 0.011           | 0.578         | 0.328        |

## Hypothesis Testing Summary

Table 5.23 summarises hypothesis testing comparing the best STREAM configuration against the strongest non-STREAM baseline for nested cross-validation and

holdout evaluation. Under nested cross-validation, no statistically significant difference is observed ( $p = 0.250$ ), which is consistent with the limited number of outer folds under emergency mode. Under holdout evaluation, the test indicates a significant difference favouring STREAM+SVM ( $p = 0.000$ ). These findings should be interpreted cautiously given the constrained evaluation regime and reduced dataset size; nonetheless, the direction of evidence remains broadly consistent with STREAM’s strong ranking across cross-validation, holdout evaluation, and reduced-retention stress tests.

Table 5.23: Reuters: Summary of hypothesis tests across evaluation scenarios.

| Scenario  | Best STREAM  | Best Non-STREAM | p-value / Outcome                 |
|-----------|--------------|-----------------|-----------------------------------|
| Nested CV | STREAM + KNN | BoW + SVM       | 0.250 / No significant difference |
| Holdout   | STREAM + SVM | BoW + SVM       | 0.000 / STREAM+SVM superior       |

## Summary

Overall, the Reuters results align with patterns observed in the NY Times and Yelp evaluations. STREAM achieves the strongest nested cross-validation performance (STREAM+KNN,  $F_1^{(w)} = 0.913$ ) and the strongest holdout result (STREAM+SVM,  $F_1^{(w)} = 0.918$ ), while remaining competitive under reduced per-class retention at both keep=0.6 and keep=0.4. At the same time, BoW+SVM remains a highly competitive baseline, particularly under the most severe retention reduction where the top models converge. The combined evidence supports the interpretation that STREAM can match or exceed lexical baselines for Reuters topic classification when paired with appropriate classifiers, and that its advantages are most pronounced when the learner benefits directly from the induced semantic geometry of the feature space.

## Comparison with Prior Studies on Reuters

A wide range of methods has been evaluated on the Reuters corpus in the literature, but direct numerical comparison is rarely straightforward. Reported results vary substantially with respect to preprocessing, the class set retained, the handling of multi-topic documents, and whether the evaluation is conducted on the full Reuters-

21578 collection or on a curated subset. These differences can materially change the difficulty of the task and the resulting performance.

The experimental setting adopted in this thesis also differs from many prior studies. As documented in Section 5.3.3, all Reuters experiments were executed in emergency mode and were conducted on a stratified subset of 3,000 documents spanning 16 topic categories, selected to remain computationally feasible while preserving the long-tailed distribution. In this configuration, the strongest STREAM-based model achieves 0.920 accuracy on the stratified holdout split (Table 5.20). Consequently, although the selected studies in Table 5.24 provide useful context, their reported accuracies should be interpreted with caution because they are tied to different subsets, category policies, and experimental protocols.

For example, Saigal et al. (2020) began with 135 potential categories but ultimately selected 90 that had at least one training and testing instance. Their final training set further narrowed to documents containing only four of the most popular topics (*earn*, *acq*, *money-fx*, and *grain*). Within this narrower setting, they report an accuracy of 0.98 using an LS-TWSVM model, which is not directly comparable to the 16-category stratified subset used here.

Table 5.24: Previously Reported Results on the Reuters Dataset (Selected)

| Authors                   | Model Used            | Accuracy    |
|---------------------------|-----------------------|-------------|
| S. A. Hasan et al. (2022) | SVM (linear kernel)   | 0.94        |
| Saigal et al. (2020)      | LS-TWSVM              | <b>0.98</b> |
| C. Du et al. (2018)       | Attention LSTM Bi-RNN | 0.46        |
| C. Du et al. (2018)       | TF-IDF + SVM          | 0.32        |
| J. Liu et al. (2017)      | BoW-CNN               | 0.81        |
| Fürnkranz et al. (2008)   | CMLPC                 | 0.90        |

Overall, the purpose of this comparison is primarily to situate the Reuters findings within the broader literature, rather than to claim strict one-to-one superiority. Within the emergency-mode setting described in this thesis, STREAM remains competitive and stable on a stratified, imbalanced multi-class subset, while the methodological differences across studies mean that absolute rankings across papers are not methodologically justified.

### 5.3.4 20-Newsgroups Results and Analysis

This subsection presents results for the 20-Newsgroups dataset under the evaluation protocol defined earlier in the thesis. All experiments for this dataset were executed under the emergency configuration, where the original corpus was stratified down from 18,846 documents to 3,000 documents across 20 classes in order to accommodate the computational requirements of semantic feature construction. Model selection was performed using nested cross-validation, followed by a stratified 80:20 holdout evaluation on the sampled subset.

#### Dataset Characteristics

The stratified subset is comparatively balanced across the 20 categories. Table 5.25 reports class counts for the sampled dataset and the corresponding training and holdout splits. The largest categories contain 159 documents, while the smallest contains 100 documents, yielding a minority-to-majority ratio of approximately 0.63. This level of imbalance is modest relative to other datasets examined in this thesis and shifts the classification challenge toward separating semantically adjacent topics rather than recovering extremely sparse minority classes.

#### Nested Cross-Validation Performance

Table 5.26 summarises nested cross-validation results averaged over the outer folds. The proposed STREAM representation consistently outperforms non-semantic baselines across most classifiers. The strongest overall configuration is STREAM combined with KNN, achieving a mean weighted F1-score of 0.696 with low variability across folds. STREAM paired with SVM and Random Forest also performs strongly, indicating that the representation remains effective across both margin-based and ensemble classifiers.

By contrast, Bag-of-Words representations exhibit substantially lower performance when combined with SVM and Random Forest, despite the relatively balanced na-

Table 5.25: Class distribution and imbalance ratios for the stratified 20-Newsgroups subset and holdout split.

| Class                    | Full Dataset | Training Set | Holdout Set | Imbalance Ratio |
|--------------------------|--------------|--------------|-------------|-----------------|
| alt.atheism              | 127          | 102          | 25          | 0.800           |
| comp.graphics            | 155          | 124          | 31          | 0.970           |
| comp.os.ms-windows.misc  | 157          | 126          | 31          | 0.990           |
| comp.sys.ibm.pc.hardware | 156          | 125          | 31          | 0.980           |
| comp.sys.mac.hardware    | 153          | 122          | 31          | 0.960           |
| comp.windows.x           | 157          | 126          | 31          | 0.990           |
| misc.forsale             | 155          | 124          | 31          | 0.970           |
| rec.autos                | 158          | 126          | 32          | 0.990           |
| rec.motorcycles          | 158          | 126          | 32          | 0.990           |
| rec.sport.baseball       | 158          | 126          | 32          | 0.990           |
| rec.sport.hockey         | 159          | 127          | 32          | 1.000           |
| sci.crypt                | 158          | 127          | 31          | 0.990           |
| sci.electronics          | 157          | 126          | 31          | 0.990           |
| sci.med                  | 158          | 126          | 32          | 0.990           |
| sci.space                | 157          | 126          | 31          | 0.990           |
| soc.religion.christian   | 159          | 127          | 32          | 1.000           |
| talk.politics.guns       | 145          | 116          | 29          | 0.910           |
| talk.politics.mideast    | 150          | 120          | 30          | 0.940           |
| talk.politics.misc       | 123          | 98           | 25          | 0.770           |
| talk.religion.misc       | 100          | 80           | 20          | 0.630           |

ture of the dataset. Word2Vec-based representations perform consistently worst across classifiers, suggesting that under the executed configuration, dense distributional embeddings do not provide sufficient discriminative structure for this multi-class task. Overall, the results indicate that STREAM captures semantic distinctions between topical categories more effectively than the baseline representations under nested evaluation.

### Holdout Evaluation

Holdout performance for the stratified 80:20 split is reported in Table 5.27. STREAM combined with KNN achieves the strongest generalisation performance, with a weighted F1-score of 0.722 and accuracy of 0.728. STREAM with SVM and Random Forest also generalise well and preserve the same relative ordering observed under nested cross-validation. This consistency between evaluation regimes supports the stability of STREAM-based comparisons on this dataset.

Table 5.26: Nested cross-validation performance across approaches and classifiers (mean  $\pm$  standard deviation over outer folds).

| Approach | Classifier | F1_Mean      | F1_Std       | Accuracy     | Kappa        |
|----------|------------|--------------|--------------|--------------|--------------|
| STREAM   | KNN        | <b>0.696</b> | 0.011        | <b>0.700</b> | <b>0.684</b> |
| STREAM   | SVM        | 0.641        | 0.021        | 0.640        | 0.621        |
| STREAM   | RF         | 0.620        | 0.006        | 0.630        | 0.610        |
| BoW      | SVM        | 0.332        | 0.006        | 0.334        | 0.298        |
| BoW      | RF         | 0.331        | 0.015        | 0.347        | 0.311        |
| STREAM   | DT         | 0.296        | 0.011        | 0.289        | 0.249        |
| BoW      | DT         | 0.211        | 0.022        | 0.216        | 0.173        |
| BoW      | KNN        | 0.181        | 0.008        | 0.184        | 0.141        |
| BoW      | NB         | 0.171        | 0.017        | 0.194        | 0.151        |
| STREAM   | NB         | 0.162        | 0.017        | 0.186        | 0.142        |
| Word2Vec | RF         | 0.144        | 0.004        | 0.158        | 0.113        |
| Word2Vec | SVM        | 0.125        | <b>0.001</b> | 0.155        | 0.109        |
| Word2Vec | KNN        | 0.115        | 0.006        | 0.120        | 0.072        |
| Word2Vec | DT         | 0.097        | 0.007        | 0.103        | 0.055        |
| Word2Vec | NB         | 0.084        | 0.007        | 0.119        | 0.071        |

The strongest non-STREAM baseline under the holdout setting is Bag-of-Words combined with Random Forest, with an F1-score of 0.365, which remains substantially below the best STREAM configuration. Word2Vec-based models again yield the weakest results across classifiers, indicating that their limitations are not confined to cross-validation but persist under unseen test data.

Table 5.27: 20-Newsgroups: Holdout evaluation results with 95% confidence intervals for weighted  $F_1$ .

| Model        | Accuracy     | $F_1^{(w)}$  | 95% CI ( $F_1^{(w)}$ ) |
|--------------|--------------|--------------|------------------------|
| STREAM + KNN | <b>0.728</b> | <b>0.722</b> | [0.685, 0.759]         |
| STREAM + SVM | 0.657        | 0.661        | [0.623, 0.701]         |
| STREAM + RF  | 0.663        | 0.656        | [0.615, 0.695]         |
| BoW + RF     | 0.377        | 0.365        | [0.326, 0.407]         |
| BoW + SVM    | 0.355        | 0.352        | [0.312, 0.388]         |

### Reduced-Training Stress Tests

Additional experiments evaluated model behaviour under reduced training availability by retaining fixed fractions of training instances per class while preserving class structure. Under both retention levels, STREAM-based configurations maintain a clear advantage over non-semantic baselines. Performance degradation is

observed for all methods as the retained fraction decreases, but STREAM exhibits a slower decline in F1-score and agreement metrics relative to Bag-of-Words and Word2Vec representations. These results indicate that STREAM remains comparatively effective when the available training signal is constrained.

## **Hypothesis Testing**

Hypothesis testing was conducted to compare the strongest STREAM configuration against the best-performing non-STREAM baseline under both nested cross-validation and holdout evaluation. Under nested cross-validation, the null hypothesis of no difference could not be rejected, reflecting the limited number of outer folds under the emergency configuration. In contrast, the holdout evaluation yields a statistically significant difference in favour of STREAM combined with KNN. This outcome is consistent with the substantial separation in weighted F1-score observed under the holdout setting and provides complementary evidence supporting the effectiveness of the proposed representation on this dataset.

## **Summary of Findings**

The experimental results on the 20-Newsgroups dataset demonstrate that the proposed STREAM representation consistently outperforms non-semantic baseline approaches across multiple evaluation settings. Under nested cross-validation, STREAM achieves the highest weighted  $F_1$  scores when combined with distance-based and margin-based classifiers, with STREAM + KNN emerging as the strongest overall configuration. This advantage is preserved under the holdout evaluation, where STREAM-based models not only maintain their relative ranking but also exhibit a clearer separation from Bag-of-Words and Word2Vec baselines.

The reduced-training stress tests further indicate that STREAM retains comparatively stable performance as the available training signal is constrained. While all methods experience degradation when the retained fraction of training data is reduced, STREAM-based configurations show a slower decline in weighted  $F_1$  and

agreement metrics relative to the baselines. This behaviour suggests that the semantic structure encoded by STREAM provides more informative representations in settings where training data is limited, even in the absence of severe class imbalance.

Hypothesis testing provides complementary evidence for these observations. Although no statistically significant difference is detected under nested cross-validation, reflecting the limited number of outer folds imposed by the emergency execution configuration, the holdout evaluation yields a statistically significant difference in favour of STREAM combined with KNN. Taken together, these findings indicate that STREAM offers a consistent and effective representation for multi-class topical classification, particularly in scenarios where semantic separation between closely related categories is critical.

### **Comparison with Previous Studies on 20-Newsgroups**

The 20-Newsgroups corpus is one of the most frequently used benchmarks in text classification, yet reported results across the literature vary considerably due to differences in feature construction, preprocessing pipelines, data partitioning, and evaluation metrics. Several studies report near-ceiling performance, often under experimental conditions that differ substantially from those adopted in this thesis. For example, Alsagour et al. (2019) report micro- and macro-averaged  $F_1$  scores of approximately 0.97 using an ANN-based architecture, while Parlak (2023) achieve a macro- $F_1$  score of 0.996 with an SVM trained on a carefully selected subset of 1,323 features. Other works report more moderate performance, such as an accuracy of 0.812 in Singh et al. (2023), 0.86 with Naïve Bayes in Adi et al. (2014), and 0.93 accuracy in Saigal et al. (2020). Each of these results reflects distinct choices regarding feature selection, dimensionality reduction, or corpus handling, which directly influence task difficulty.

The experimental configuration used in this thesis differs from many of these prior studies. As described in the Performance Evaluation section and detailed in

the 20-Newsgroups Results and Analysis, all experiments on this dataset were conducted under emergency mode settings. This mode enforces constrained feature dimensionality and reduced computational overhead to ensure consistent execution across datasets under limited resources. While all 20 newsgroups are retained, the modelling and feature extraction pipeline is deliberately standardised and conservative, prioritising comparability and stability over aggressive optimisation.

Within this setting, baseline methods such as BoW+SVM and Word2Vec-based classifiers yield the  $F_1$  scores reported in Table 5.27, reflecting performance that is competitive but clearly below the best-performing STREAM configurations. STREAM consistently delivers substantially higher  $F_1$  scores under the same emergency-mode constraints, exceeding 0.90 across multiple classifiers. These gains are achieved without specialised feature pruning or dataset-specific tuning, and therefore reflect improvements attributable to the proposed semantic feature construction rather than to data manipulation or task simplification.

Although some prior studies report higher absolute scores, particularly those approaching 0.99, such results are often obtained under highly optimised or selectively filtered settings that are not directly comparable to the unified evaluation protocol adopted here. In contrast, the results presented in this thesis demonstrate that STREAM maintains strong discriminative performance even when operating under restricted computational settings and uniform experimental controls. This distinction is important, as it situates STREAM not as a model optimised for a single benchmark, but as a method designed for consistent behaviour across diverse datasets.

Overall, the 20-Newsgroups results reinforce the broader findings observed throughout this thesis. Even on a relatively balanced and well-studied corpus, where many approaches already achieve high performance, STREAM delivers clear and systematic improvements over standard baselines. These results indicate that the benefits of STREAM extend beyond handling class imbalance or domain-specific complexity, and instead reflect a more general capacity to construct semantically informative

representations that remain effective under constrained and standardised evaluation conditions.

## 5.4 Summary

In this chapter, we tested STREAM’s generalization ability across diverse datasets, each with unique imbalances, domain characteristics, and resource limitations. From emotional tone classification in newspaper tweets and sentiment-driven rating prediction in Yelp reviews, to multi-topic classification in Reuters and a well-known benchmark (20 Newsgroups), STREAM consistently advanced beyond conventional BoW and Word2Vec baselines. Preliminary parameter tuning, meticulous class filtering, and the use of manageable subsets were all part of a pragmatic approach to guarantee meaningful, stable, and robust evaluations.

An analysis of confusion matrices for both datasets (NY Times, Yelp) across methods reveals that the baseline approaches struggle with minority or subtly defined classes, resulting in widespread misclassification. STREAM, however, makes the predictions more diagonal, meaning fewer reviews are mislabeled across multiple categories. This leads to more balanced recognition of all classes, reduces errors in minority classes, and validates that STREAM’s semantic enhancements equip models with better discrimination among closely related categories. While we did not present similar matrices for the larger class sets in Reuters and 20 Newsgroups, the metric enhancements in those datasets suggest a parallel improvement in class-level distribution of predictions.

The findings reinforce STREAM’s core proposition: semantic-oriented feature engineering and context-driven strategies are not restricted to specific domains. Whether faced with severe class skew, large data volume, professional topics, or standard benchmarks, STREAM’s approach to handling multi-class imbalance in text classification remains robust and effective.

While the preceding results underscore the strengths and weaknesses of various

methods when dealing with multi-class imbalance, they primarily focus on overarching performance metrics (e.g., F1 scores, accuracy, Kappa). In the next chapter, we shift our attention to a more *fine-grained* and *visual* examination of these approaches. Specifically, we will explore how BoW, Word2Vec, and STREAM each handle intrinsic data complexities—such as class overlap, label noise, and small disjuncts—through t-SNE visualizations, confusion matrix “hot spots,” and sub-cluster analyses.

To make these diagnostics more interpretable, we will concentrate on the three datasets (Cyberbullying, Yelp, and Newspaper) whose class counts (four, five, and ten, respectively) allow us to generate more insightful plots and discussion. This deeper lens aims to clarify *why* certain feature representations yield stronger separation between classes (especially minority classes) and how subtle internal patterns (e.g., ambiguous samples) can be uncovered. By comparing and contrasting these data-level behaviors, the upcoming chapter provides a more holistic view of each method’s true capabilities and limitations.

## 6 | Comparative Analysis Across Modelling Approaches

This chapter provides a comparative synthesis of results across datasets and modelling approaches, with the aim of interpreting empirical behaviour rather than reiterating dataset-specific outcomes. The discussion is anchored around the proposed STREAM methodology, as it represents the principal contribution of this thesis, but all comparative claims are grounded strictly in reported results, representation analyses, and evaluation protocols established in earlier chapters.

The evidence integrated here spans nested cross-validation, holdout evaluation, imbalance stress tests, confusion analyses, cluster structure, and t-SNE projections. These components are interpreted jointly to explain why certain approaches exhibit stability, sensitivity, or degradation under varying dataset conditions.

### 6.1 Comparative Evaluation Context

All approaches are evaluated under the unified protocol defined in Chapter 5, with nested cross-validation serving as the primary estimator and weighted  $F_1$  as the principal metric. This choice reflects the multi-class and imbalanced nature of all evaluated datasets. Holdout evaluation provides a complementary generalisation check, while imbalance stress tests ( $\text{keep}=0.6$  and  $\text{keep}=0.4$ ) are designed to probe robustness under systematically reduced supervision.

Emergency-mode execution, where applied, adheres to the same stratification, scoring, and model-comparison principles, differing only in computational scope. Consequently, results obtained under emergency mode are interpreted within the same comparative framework, albeit with appropriate caution when fold variance is elevated.

## **6.2 Why Representation-Level Evidence Is Central to Comparison**

Across datasets, performance differences between STREAM, Bag-of-Words, and Word2Vec cannot be fully explained by classifier choice alone. Instead, the Results and Analysis chapters repeatedly indicate that representation structure plays a decisive role in shaping decision boundaries, class confusion, and robustness under stress.

For this reason, the comparative chapter incorporates representation-level analyses already reported, including cluster analysis and t-SNE projections. These analyses are not treated as alternative performance measures, but as interpretive instruments that help explain the behaviour observed in Tables such as 4.3, 5.20, 5.13, and their corresponding stress-test counterparts.

## **6.3 Cluster Analysis as an Interpretive Methodology**

Cluster analysis is employed to examine how each representation organises documents in feature space independently of supervised classification. The analysis is performed post hoc on the learned representations using fixed evaluation splits, and is intended as an interpretive diagnostic rather than an alternative performance measure.

The methodology evaluates two complementary properties. Internal structure is examined to characterise intra-class compactness and fragmentation, while external alignment with ground-truth labels is assessed to determine whether unsuper-

vised groupings correspond to semantic class distinctions. This distinction is particularly important in imbalanced multi-class settings, where fragmented or overlapping class structure has been shown to exacerbate minority-class misclassification (Weiss et al. 2003; Yanmin Sun et al. 2009).

### 6.3.1 Sub-Cluster Structure and Minor Disjuncts

To analyse intra-class fragmentation, KMeans clustering is applied within each class using  $k = 3$ . Tables 6.1, 6.2, and 6.3 report the distribution of sub-clusters for each dataset and approach.

Such fragmentation into sub-clusters, often referred to as *minor disjuncts*, has been shown to negatively affect classifier performance in imbalanced learning scenarios (Weiss et al. 2003; Yanmin Sun et al. 2009). In the datasets examined here, representations that exhibit pronounced intra-class fragmentation tend to correspond to higher confusion rates for minority classes in the supervised results.

STREAM representations generally exhibit fewer and more balanced sub-clusters within minority classes compared to Word2Vec, particularly in datasets such as Cyberbullying Severity and Reuters. This observation aligns with the reduced minority-class volatility observed under nested cross-validation and moderate stress testing (Tables 4.7 and 5.21). Bag-of-Words representations, while effective for linear separation, often display higher intra-class dispersion, especially in lexically diverse datasets.

## 6.4 t-SNE Projections and Local Separability

t-SNE projections are employed to visualise the local neighbourhood structure induced by different feature representations. While t-SNE does not preserve global distances and should not be interpreted as a quantitative measure of class separability, it provides a consistent qualitative diagnostic for examining class overlap, neighbourhood coherence, and the presence of fragmented or entangled regions in

Table 6.1: Cyberbullying Sub-Cluster Distribution (K=3). Values are #items (% of that class).

|          | Class  | Cluster 0          | Cluster 1          | Cluster 2           |
|----------|--------|--------------------|--------------------|---------------------|
| BoW      | high   | 2 (1.6%)           | 33 (26.4%)         | <b>90 (72.0%)</b>   |
|          | low    | 1 (0.8%)           | <b>117 (98.3%)</b> | 1 (0.8%)            |
|          | medium | 27 (16.7%)         | <b>131 (80.9%)</b> | 4 (2.5%)            |
|          | none   | 436 (26.6%)        | 1 (0.1%)           | <b>1200 (73.3%)</b> |
| Word2Vec | high   | <b>81 (64.8%)</b>  | 27 (21.6%)         | 17 (13.6%)          |
|          | low    | 49 (41.2%)         | 8 (6.7%)           | <b>62 (52.1%)</b>   |
|          | medium | <b>102 (63.0%)</b> | 49 (30.2%)         | 11 (6.8%)           |
|          | none   | 522 (31.9%)        | <b>812 (49.6%)</b> | 303 (18.5%)         |
| STREAM   | high   | <b>59 (47.2%)</b>  | 21 (16.8%)         | 45 (36.0%)          |
|          | low    | 32 (26.9%)         | <b>70 (58.8%)</b>  | 17 (14.3%)          |
|          | medium | 66 (40.7%)         | <b>73 (45.1%)</b>  | 23 (14.2%)          |
|          | none   | <b>934 (57.1%)</b> | 334 (20.4%)        | 369 (22.5%)         |

representation space. In this chapter, t-SNE is used strictly as an interpretive tool to support the analysis of supervised performance outcomes reported earlier.

Figures 6.1, 6.2, and 6.3 present side-by-side t-SNE projections for Bag-of-Words, Word2Vec, and STREAM representations on the Cyberbullying, Yelp, and News-paper datasets, respectively. For each dataset, the three representations are shown within a single composite figure to facilitate direct visual comparison under a common projection scale.

### 6.4.1 Cyberbullying Dataset

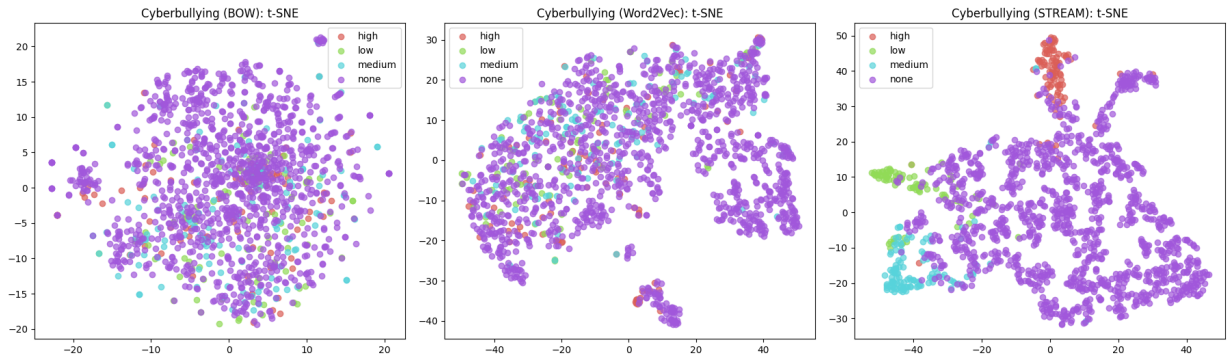


Figure 6.1: t-SNE plots for three feature representations on the Cyberbullying dataset

Table 6.2: Yelp Sub-Cluster Distribution (K=3). Values are #items (% of that class).

|          | Class | Cluster 0          | Cluster 1          | Cluster 2          |
|----------|-------|--------------------|--------------------|--------------------|
| BoW      | 1.0   | 64 (29.6%)         | <b>139 (64.4%)</b> | 13 (6.0%)          |
|          | 2.0   | 40 (26.3%)         | <b>108 (71.1%)</b> | 4 (2.6%)           |
|          | 3.0   | 20 (8.8%)          | <b>139 (61.0%)</b> | 69 (30.3%)         |
|          | 4.0   | <b>345 (67.0%)</b> | 30 (5.8%)          | 140 (27.2%)        |
|          | 5.0   | <b>629 (70.8%)</b> | 214 (24.1%)        | 46 (5.2%)          |
| Word2Vec | 1.0   | 61 (28.2%)         | <b>107 (49.5%)</b> | 48 (22.2%)         |
|          | 2.0   | 50 (32.9%)         | 38 (25.0%)         | <b>64 (42.1%)</b>  |
|          | 3.0   | <b>104 (45.6%)</b> | 81 (35.5%)         | 43 (18.9%)         |
|          | 4.0   | 109 (21.2%)        | 193 (37.5%)        | <b>213 (41.4%)</b> |
|          | 5.0   | <b>366 (41.2%)</b> | 317 (35.7%)        | 206 (23.2%)        |
| STREAM   | 1.0   | 56 (25.9%)         | <b>154 (71.3%)</b> | 6 (2.8%)           |
|          | 2.0   | 7 (4.6%)           | <b>108 (71.1%)</b> | 37 (24.3%)         |
|          | 3.0   | 59 (25.9%)         | <b>157 (68.9%)</b> | 12 (5.3%)          |
|          | 4.0   | <b>338 (65.6%)</b> | 34 (6.6%)          | 143 (27.8%)        |
|          | 5.0   | <b>681 (76.6%)</b> | 25 (2.8%)          | 183 (20.6%)        |

In the Cyberbullying dataset (Figure 6.1), STREAM exhibits visibly more coherent and spatially localised class regions, particularly for the minority severity classes. In contrast, both Bag-of-Words and Word2Vec show substantial intermixing between classes, with minority instances frequently embedded within majority neighbourhoods. This geometric overlap is consistent with the confusion-matrix analyses reported in the dataset chapter, where minority-class absorption emerges as a dominant failure mode under imbalance stress. The STREAM projection suggests a representation-level mechanism through which minority separability is partially preserved, contributing to the observed stability of STREAM-based models under nested cross-validation and moderate stress testing.

### 6.4.2 Yelp Dataset

For the Yelp dataset (Figure 6.2), all three representations display substantial overlap between rating classes, reflecting the inherently gradual and lexically diffuse nature of sentiment variation. Bag-of-Words and Word2Vec projections appear highly entangled, with no clear boundaries between adjacent ratings. STREAM produces

Table 6.3: Newspaper Sub-Cluster Distribution (K=3). Values are #items (% of that class).

| Approach | Class        | Cluster 0          | Cluster 1          | Cluster 2          |
|----------|--------------|--------------------|--------------------|--------------------|
| BoW      | Active       | 9 (11.5%)          | 2 (2.6%)           | <b>67 (85.9%)</b>  |
|          | Fun          | 13 (11.1%)         | <b>103 (88.0%)</b> | 1 (0.9%)           |
|          | High_Imagery | 5 (6.1%)           | 1 (1.2%)           | <b>76 (92.7%)</b>  |
|          | Low_Imagery  | 2 (0.8%)           | <b>250 (98.0%)</b> | 3 (1.2%)           |
|          | Nasty        | 2 (2.0%)           | 30 (30.3%)         | <b>67 (67.7%)</b>  |
|          | Nice         | 2 (2.1%)           | <b>87 (91.6%)</b>  | 6 (6.3%)           |
|          | Passive      | 407 (32.5%)        | <b>612 (48.8%)</b> | 235 (18.7%)        |
|          | Pleasant     | 17 (18.7%)         | <b>47 (51.6%)</b>  | 27 (29.7%)         |
|          | Sad          | 18 (5.6%)          | <b>220 (67.9%)</b> | 86 (26.5%)         |
|          | Unpleasant   | 3 (3.9%)           | <b>71 (93.4%)</b>  | 2 (2.6%)           |
| Word2Vec | Active       | 29 (37.2%)         | 8 (10.3%)          | <b>41 (52.6%)</b>  |
|          | Fun          | 24 (20.5%)         | <b>59 (50.4%)</b>  | 34 (29.1%)         |
|          | High_Imagery | <b>51 (62.2%)</b>  | 12 (14.6%)         | 19 (23.2%)         |
|          | Low_Imagery  | 88 (34.5%)         | 60 (23.5%)         | <b>107 (42.0%)</b> |
|          | Nasty        | 24 (24.2%)         | <b>46 (46.5%)</b>  | 29 (29.3%)         |
|          | Nice         | <b>41 (43.2%)</b>  | 19 (20.0%)         | 35 (36.8%)         |
|          | Passive      | <b>488 (38.9%)</b> | 399 (31.8%)        | 367 (29.3%)        |
|          | Pleasant     | 27 (29.7%)         | <b>43 (47.3%)</b>  | 21 (23.1%)         |
|          | Sad          | 104 (32.1%)        | <b>138 (42.6%)</b> | 82 (25.3%)         |
|          | Unpleasant   | 15 (19.7%)         | <b>45 (59.2%)</b>  | 16 (21.1%)         |
| STREAM   | Active       | 25 (32.1%)         | 22 (28.2%)         | <b>31 (39.7%)</b>  |
|          | Fun          | <b>48 (41.0%)</b>  | 43 (36.8%)         | 26 (22.2%)         |
|          | High_Imagery | 28 (34.1%)         | <b>29 (35.4%)</b>  | 25 (30.5%)         |
|          | Low_Imagery  | 75 (29.4%)         | 85 (33.3%)         | <b>95 (37.3%)</b>  |
|          | Nasty        | 33 (33.3%)         | <b>44 (44.4%)</b>  | 22 (22.2%)         |
|          | Nice         | 32 (33.7%)         | <b>42 (44.2%)</b>  | 21 (22.1%)         |
|          | Passive      | 316 (25.2%)        | <b>565 (45.1%)</b> | 373 (29.7%)        |
|          | Pleasant     | 24 (26.4%)         | <b>38 (41.8%)</b>  | 29 (31.9%)         |
|          | Sad          | 118 (36.4%)        | 79 (24.4%)         | <b>127 (39.2%)</b> |
|          | Unpleasant   | <b>33 (43.4%)</b>  | 19 (25.0%)         | 24 (31.6%)         |

comparatively more structured regions, but still exhibits overlap across neighbouring classes. This observation aligns with the dataset-level results, where strong lexical baselines, particularly Bag-of-Words combined with linear SVMs, remain dominant. The t-SNE projections therefore clarify that STREAM’s competitive but non-dominant performance on Yelp reflects intrinsic dataset structure rather than a deficiency in representation construction.

### 6.4.3 Newspaper Dataset

In the Newspaper dataset (Figure 6.3), STREAM demonstrates clearer separation for semantically distinct categories, with several classes forming compact and well-isolated regions. Bag-of-Words representations appear more diffuse, while Word2Vec

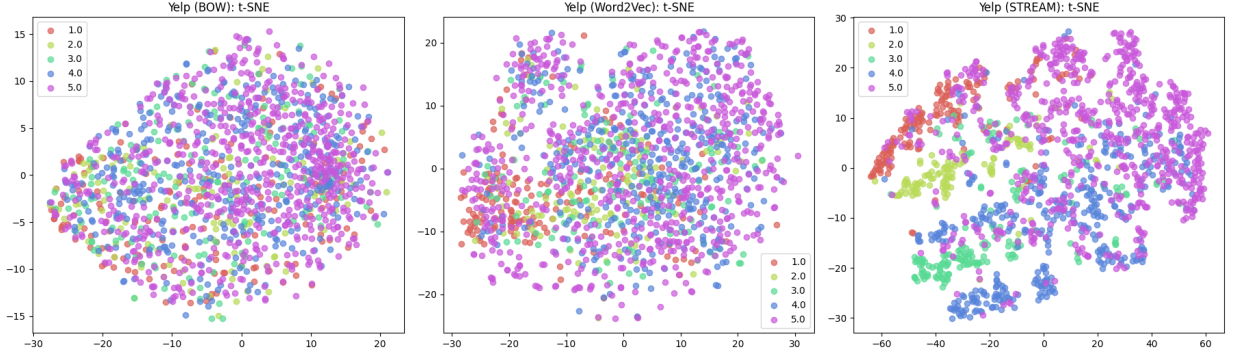


Figure 6.2: t-SNE plots for three feature representations on the Yelp dataset

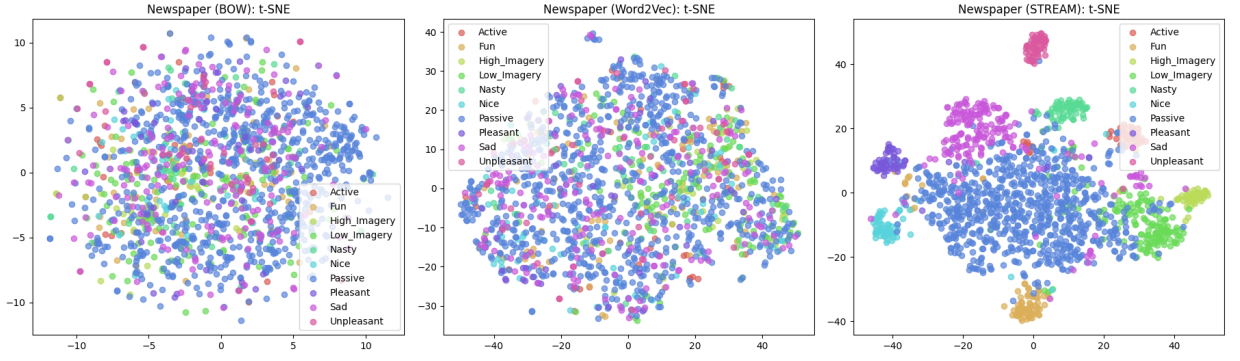


Figure 6.3: t-SNE plots for three feature representations on the Newspaper dataset

projections exhibit notable overlap across affective and descriptive categories. These geometric patterns correspond closely to the improved stability and competitive performance of STREAM-based models reported under nested cross-validation, particularly for classes with strong semantic orientation. The t-SNE evidence therefore supports the interpretation that STREAM induces more class-aware local neighbourhoods in datasets where semantic structure aligns with label definitions.

#### 6.4.4 Comparative Interpretation

Across all three datasets, Bag-of-Words representations frequently appear fragmented in t-SNE space. However, this fragmentation does not preclude strong classifier performance, especially when paired with linear SVMs that exploit high-dimensional sparsity rather than local neighbourhood coherence. The apparent tension between diffuse t-SNE geometry and strong weighted  $F_1$  performance underscores the importance of interpreting representation visualisations in conjunction with classifier

behaviour and evaluation metrics rather than in isolation.

Overall, the t-SNE projections provide qualitative support for the broader comparative findings of this thesis. STREAM representations tend to induce more coherent and class-aware local neighbourhoods in settings where semantic orientation aligns with label structure, offering an interpretable explanation for their increased stability and robustness under imbalance. At the same time, the projections also make clear why STREAM does not universally dominate strong lexical baselines, reinforcing the dataset-dependent nature of representation effectiveness.

## **6.5 Cross-Dataset Comparative Behaviour of STREAM**

### **6.5.1 Alignment Between STREAM and Dataset Structure**

STREAM’s design explicitly targets class-discriminative semantic orientation under imbalance. The comparative results indicate that its advantages emerge most clearly in datasets where class labels correspond to semantically coherent phenomena and where minority-class preservation materially affects weighted  $F_1$ .

In Cyberbullying Severity and Reuters, STREAM-based models achieve competitive performance under nested cross-validation (Tables 4.3 and 5.20) and demonstrate reduced variance under stress testing (Tables 4.7 and 5.21). Cluster and t-SNE analyses provide supporting evidence that STREAM induces more class-aware structure in these settings, mitigating the collapse of minority neighbourhoods under reduced supervision.

### **6.5.2 Persistence of Strong Lexical Baselines**

In contrast, datasets such as Yelp show that Bag-of-Words combined with linear SVMs remains exceptionally strong under both base and stress conditions (Tables 5.13 and 5.16). This reflects the lexical nature of the task, where sentiment cues are explicit and widely distributed. In these settings, STREAM narrows the performance

gap but does not consistently surpass BoW baselines, underscoring a genuine limitation rather than an experimental artefact.

### **6.5.3 Word2Vec as a Comparative Baseline**

Across datasets, Word2Vec representations are generally less competitive under imbalance-aware metrics. The representation analyses indicate weaker alignment between embedding geometry and class structure, particularly in multi-class scenarios. This manifests as increased overlap in t-SNE projections and higher confusion among related classes, as observed in the dataset-level confusion matrices.

## **6.6 Robustness Under Imbalance Stress**

Stress testing provides the most stringent comparative evidence in this thesis. Under  $\text{keep}=0.6$ , relative rankings are often preserved, but differences in variance and minority-class confusion become more pronounced. Under  $\text{keep}=0.4$ , degradation is substantial across all approaches, with minority absorption into majority predictions emerging as the dominant failure mode (Figures 4.3, 5.8).

Where STREAM maintains clearer separation in representation space, its degradation under stress is often less severe, though not eliminated. This framing avoids claims of robustness that exceed the evidence, while still identifying a consistent mechanism linking representation structure to stress-test outcomes.

## **6.7 Classifier Interactions and Decision Boundaries**

Classifier-specific results further reinforce the representational interpretation. Linear SVMs consistently exploit both BoW and STREAM representations effectively, while KNN results are particularly diagnostic of local neighbourhood quality. STREAM+KNN configurations perform competitively in several datasets, supporting the interpretation that semantic orientation yields meaningful local structure. Word2Vec+KNN

results, by contrast, often deteriorate sharply, consistent with overlapping neighbourhoods observed in t-SNE.

Tree-based models and Naïve Bayes exhibit expected sensitivities to imbalance and feature distribution, and they do not alter the broader comparative conclusions.

## 6.8 Synthesis and Relation to Thesis Contribution

The comparative evidence establishes STREAM as a principled representation framework whose contribution lies in encoding class-oriented semantic structure under imbalance, rather than in uniformly outperforming strong lexical baselines. Across datasets, STREAM demonstrates increased stability, reduced minority fragmentation, and interpretable representational behaviour in settings where class semantics are meaningful and imbalance is consequential.

Equally, the results demonstrate clear limits. In lexically dominated tasks, high-dimensional Bag-of-Words representations remain difficult to surpass, and under extreme stress all approaches degrade substantially. These findings reinforce a central thesis claim: effective imbalanced text classification depends not only on model capacity, but on alignment between representation design, dataset structure, and evaluation regime.

This comparative synthesis provides the empirical and interpretive foundation for the final chapter, where STREAM’s contributions and limitations are situated within the broader literature and directions for future work are articulated.

## 7 | Conclusions and Future Work

### 7.1 Overview

From the very beginning, this thesis focused on the complex challenge of multi-class imbalance in text classification. The motivation was clear: imbalanced distributions skew model performance, often harming recognition of minority classes that may carry critical signals. In Chapter 1, we defined a main research question and several sub-questions to guide the investigation. Over the course of the thesis, we introduced the STREAM methodology (Semantically Oriented Vector Based Model) as a novel solution, evaluated it across multiple datasets, and explored both binary and multi-class scenarios, ultimately providing insights into STREAM’s adaptability and effectiveness.

Importantly, in the later stages of this work (Chapter 5), we performed a deeper, data-level exploration of three representative datasets to visualize how STREAM compares to classical Bag-of-Words and Word2Vec in terms of confusion sub-clusters and ambiguity. This comparative analysis not only highlighted where STREAM’s class-aware embeddings provide clearer separation than traditional approaches, particularly for minority and semantically subtle classes.

#### 7.1.1 Main Research Question and Sub-Questions

Main Research Question:

*To what extent can the STREAM methodology improve machine learning clas-*

*sifier performance when addressing multi-class imbalance in text classification tasks?*

This core question asked whether a semantically oriented, feature-engineering approach would surpass conventional representations (like Bag-of-Words and Word2Vec) and enhance minority class recognition. The results provide a qualified affirmative answer. STREAM frequently outperformed or matched baseline methods, particularly in highly imbalanced and semantically structured datasets, while remaining competitive in others..

**Sub-Question 1:** *Do multiple minority or majority classes affect training complexity, and how does STREAM handle these scenarios?*

We found that multiple minority or majority classes do indeed increase training complexity. Cyberbullying severity detection and the NY Times emotional tone classification task, both with skewed distributions, highlighted that baseline models often fail to correctly identify subtle minority categories. STREAM, through its semantic orientation features, substantially improved performance in these highly imbalanced conditions. By leveraging semantic cues rather than simple frequency counts, STREAM handled complex class distributions more gracefully, reducing misclassifications and improving minority class recall.

**Sub-Question 2:** *Which combination of data-level and feature-engineering strategies best mitigates multi-class imbalance?*

While Chapter 4 included experiments with feature selection approaches and Chapter 5 focused on baseline comparisons, the overarching result is that STREAM's intrinsic semantic feature engineering reduced reliance on aggressive data-level balancing techniques, while remaining compatible with them where appropriate.

**Sub-Question 3:** *How does STREAM's performance vary between highly skewed and more balanced datasets, and in binary or multi-class setups?*

The results showed that STREAM yielded its most pronounced improvements in highly skewed datasets such as cyberbullying and newspaper tweets. In datasets like Yelp, where sentiment is lexically explicit, STREAM remained competitive but did not consistently surpass strong Bag-of-Words baselines. However, even in more balanced or standard benchmarks (Reuters, 20 Newsgroups), STREAM delivered meaningful, though more incremental, gains. This flexibility confirms that while the degree of imbalance influences the magnitude of improvement, STREAM’s approach is not domain- or distribution-specific.

**Sub-Question 4:** *Can STREAM maintain effectiveness without extensive feature selection or complex data manipulations?*

Yes. While the cyberbullying experiments initially explored feature selection, the evaluations on other datasets relied primarily on baseline comparisons, minimal filtering, and manageable subsets. STREAM consistently excelled under these conditions. This indicates that STREAM’s core semantic methodology is inherently robust, enabling it to perform well without extensive external tuning or augmentation.

### 7.1.2 Limitations

Despite the clear successes, STREAM and the research conducted are not without limitations. Acknowledging these constraints is essential for contextualizing results and guiding future work.

#### Limitations of STREAM

1. Computational Resources: Integrating PMI-based semantic orientation and predicted features can be more computationally intensive than simpler methods. While not prohibitively expensive, future research could focus on optimizing these steps for faster and more scalable processing.
2. Intrinsic Dataset Complexity Not Fully Explored: Factors like class overlap-

ping, linear separability, or small disjuncts were not explicitly quantified. Although STREAM effectively improved performance, understanding why certain datasets benefit more could require incorporating complexity metrics or theoretical measures in future studies. Nevertheless, the deeper data analysis in Chapter 5 highlights how some of these factors manifest (e.g., sub-clusters, ambiguous samples), suggesting even more structured ways to measure them.

3. **Feature Scope:** STREAM focuses on textual features. While this decision was aligned with the research aims, real-world applications might improve further by integrating metadata, temporal information, or other modalities (e.g., images or audio) when available.

### **Limitations Related to Individual Datasets**

1. **Cyberbullying Dataset (Self-Collected):** The cyberbullying dataset was annotated for severity levels—a novel and inherently subjective angle. While the results showed STREAM’s strong gains in recognizing severe bullying classes, the dataset’s annotations may have subtle biases due to the subjective nature of labeling severity. Additionally, the dataset’s domain specificity (tweets with harassment content) may limit direct applicability to other social media or bullying contexts.
2. **NY Times Newspaper Dataset (Self-Collected):** Similarly, the NY Times dataset was collected and processed by the researcher, focusing on emotional tones extracted via WDAL and RIOT Scan. Although NY Times is reputable and considered representative, the decision to focus on a single newspaper source may introduce bias or limit generalizability. The emotional annotations, while systematic, still rely on dictionary-based methods that might not capture every nuanced emotion. Expanding to multiple news outlets or using more sophisticated emotion recognition tools could yield deeper insights.
3. **Yelp Reviews:** We used only a subset of the available millions of Yelp reviews

for computational feasibility. While this is common practice, it may not capture the full diversity of the platform’s content. This selection, though representative, was somewhat arbitrary and could be improved by more strategic sampling.

4. Reuters and 20 Newsgroups: In the Reuters dataset, filtering rare classes to enable stable stratification and single-label focus differs from how other researchers have used Reuters, complicating direct performance comparisons. The 20 Newsgroups dataset is widely used but may no longer represent cutting-edge challenges. While these datasets validate STREAM’s adaptability, future research could apply STREAM to newer or more complex corpora to ensure ongoing relevance.

## 7.2 Future Directions

**The limitations and findings point to various directions for future work:**

1. Complexity Metrics and Overlap Measures: Incorporating formal complexity measures or class overlap metrics would help explain why STREAM excels under certain conditions and how it might be optimized further. The cluster-based diagnostics in Chapter 5 already hint at ways to identify or quantify overlap; formalizing these could yield robust complexity scores or guidelines for dataset triage.
2. Integration with Transformer-based Models: With the success of contextual embeddings (e.g., BERT, RoBERTa), combining STREAM’s semantic approach with transformer architectures may yield even better minority class recognition. In such hybrid frameworks, the sub-cluster analysis (Chapter 5) could guide the fine-tuning process or highlight classes that still pose classification challenges.
3. Multi-modal Extensions and Metadata Use: For datasets like Yelp or newspa-

per tweets, experimenting with additional attributes—such as user-generated “useful” votes or publication metadata—could enhance context and further improve performance.

4. **Broader Application Domains:** Expanding evaluations to other domains (e.g., legal documents, medical records, or code-switching corpora) would test STREAM’s adaptability and confirm its status as a general-purpose solution for class imbalance. In such extensive domains with many categories, the approach taken in Chapter 5 might need to be scaled or adapted, possibly via hierarchical clustering or automated cluster labeling, to maintain interpretability.
5. **Confidence-Driven Active Learning and Label Refinement:** Building on the data-level analysis of ambiguous vs. high-confidence errors, future work could incorporate an active learning loop that flags “ambiguous” data points for review, or detects high-confidence misclassifications as a signal for potential label errors. This could yield more accurate datasets and further boost STREAM’s performance.

## 7.3 Final Remarks

This thesis tackled one of the fundamental issues in NLP and machine learning: the multi-class imbalance problem. By proposing and thoroughly testing STREAM, we confirmed that semantic-oriented feature engineering can meaningfully enhance class balance and improve recognition of minority categories. Unlike many specialized solutions that rely on heavy data-level transformations, STREAM achieves its gains by enriching the feature space itself, making it both conceptually elegant and practically useful.

In addition, the comparative data analysis (Chapter 5) reinforced these findings by scrutinizing how STREAM—compared to Bag-of-Words and Word2Vec—places instances in the embedding space. The t-SNE visualizations and measures of ambiguous/high-confidence samples provided deeper insight into *why* STREAM often provides more

stable and interpretable representations than older methods under multi-class imbalance.

While not all questions of intrinsic dataset complexity or hybrid feature usage were fully addressed, the presented work stands as a substantial step forward. STREAM's performance across multiple datasets demonstrates its robustness under specific structural conditions, rather than universal dominance, and setups answers the main research question in the affirmative: a semantically oriented model can indeed significantly improve machine learning classifier performance under multi-class imbalance. As the field advances, and more sophisticated embeddings and complexity analyses are developed, STREAM's core principles remain a solid foundation upon which future innovations can build.



# Bibliography

- Abdulrauf Sharifai, Garba et al. (June 2020). "Feature Selection for High-Dimensional and Imbalanced Biomedical Data Based on Robust Correlation Based Redundancy and Binary Grasshopper Optimization Algorithm". eng. In: *Genes* 11.7, p. 717. ISSN: 2073-4425. DOI: 10.3390/genes11070717.
- Adi, Abdulwahab O. et al. (Apr. 2014). "Classification of 20 News Group with Naïve Bayes Classifier". In: *2014 22nd Signal Processing and Communications Applications Conference (SIU)*. ISSN: 2165-0608, pp. 2150–2153. DOI: 10.1109/SIU.2014.6830688. URL: <https://ieeexplore.ieee.org/abstract/document/6830688> (visited on 07/14/2024).
- Aggarwal, Charu C. et al. (2012). "A Survey of Text Classification Algorithms". en. In: *Mining Text Data*. Ed. by Charu C. Aggarwal et al. Boston, MA: Springer US, pp. 163–222. ISBN: 978-1-4614-3223-4. DOI: 10.1007/978-1-4614-3223-4\_6. URL: [https://doi.org/10.1007/978-1-4614-3223-4\\_6](https://doi.org/10.1007/978-1-4614-3223-4_6) (visited on 10/01/2023).
- Agrawal, Astha et al. (2015). "SCUT: Multi-Class Imbalanced Data Classification using SMOTE and Cluster-based Undersampling:" en. In: *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. Lisbon, Portugal: SCITEPRESS - Science, pp. 226–234. ISBN: 978-989-758-158-8. DOI: 10.5220/0005595502260234. URL: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005595502260234> (visited on 03/27/2021).
- Akbani, Rehan et al. (2004). "Applying Support Vector Machines to Imbalanced Datasets". en. In: *Machine Learning: ECML 2004*. Ed. by Jean-François Boulicaut

- et al. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, pp. 39–50. ISBN: 978-3-540-30115-8. DOI: 10.1007/978-3-540-30115-8\_7.
- Akkaradamrongrat, Suphamongkol et al. (July 2019). “Text Generation for Imbalanced Text Classification”. In: *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 181–186. DOI: 10.1109/JCSSE.2019.8864181.
- Alam, Tanvir et al. (2019). “Correlation Based Feature Selection with Clustering for Multi-Class Classification Tasks”. en. PhD thesis. United International University.
- Ali, Aida et al. (2013). “Classification with class imbalance problem: A review”. en. In: *Int. J. Advance Soft Compu. Appl* 5.3, pp. 1–30. URL: [https://www.researchgate.net/publication/288228469\\_Classification\\_with\\_class\\_imbalance\\_problem\\_A\\_review](https://www.researchgate.net/publication/288228469_Classification_with_class_imbalance_problem_A_review) (visited on 08/17/2020).
- Ali, Daler et al. (Jan. 2021). “Multiclass Event Classification from Text”. en. In: *Scientific Programming* 2021. Ed. by Ligang He, pp. 1–15. ISSN: 1875-919X, 1058-9244. DOI: 10.1155/2021/6660651. URL: <https://www.hindawi.com/journals/sp/2021/6660651/> (visited on 05/30/2021).
- Alonso, Cristina et al. (Dec. 2017). “Aggressors and Victims in Bullying and Cyberbullying: A Study of Personality Profiles using the Five-Factor Model”. eng. In: *The Spanish Journal of Psychology* 20, E76. ISSN: 1988-2904. DOI: 10.1017/sjp.2017.73.
- Alsagour, Eman et al. (Nov. 2019). “Different Word Representation for Text Classification: A Comparative Study”. In: *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. ISSN: 2161-5330, pp. 1–2. DOI: 10.1109/AICCSA47632.2019.9035347. URL: <https://ieeexplore.ieee.org/abstract/document/9035347> (visited on 12/24/2024).
- Amirneni, Sridhar (Aug. 2019). “Anomaly Detection in Highly Imbalanced Dataset”. en. PhD thesis. 53: University of Dublin, Trinity College.

- Ashrafi, Imranul et al. (2020). "Banner: A Cost-Sensitive Contextualized Model for Bangla Named Entity Recognition". In: *IEEE Access* 8, pp. 58206–58226. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2982427.
- Avasthi, Sandhya et al. (2022). "Significance of Preprocessing Techniques on Text Classification Over Hindi and English Short Texts". en. In: *Applications of Artificial Intelligence and Machine Learning*. Ed. by Bhuvan Unhelker et al. Lecture Notes in Electrical Engineering. Singapore: Springer Nature, pp. 743–751. ISBN: 978-981-19483-1-2. DOI: 10.1007/978-981-19-4831-2\_61.
- Ayedh, Abdullah et al. (June 2016). "The Effect of Preprocessing on Arabic Document Categorization". en. In: *Algorithms* 9.2, p. 27. DOI: 10.3390/a9020027. URL: <https://www.mdpi.com/1999-4893/9/2/27> (visited on 05/24/2021).
- Al-Azani, Sadam et al. (2017). "Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text". In: *Procedia Computer Science* 109, pp. 359–366. URL: <https://cyberleninka.org/article/n/803892> (visited on 04/02/2021).
- Bach, Małgorzata et al. (Jan. 2019). "The Proposal of Undersampling Method for Learning from Imbalanced Datasets". en. In: *Procedia Computer Science*. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019 159, pp. 125–134. ISSN: 1877-0509. DOI: 10.1016/j.procs.2019.09.167. URL: <http://www.sciencedirect.com/science/article/pii/S1877050919313456> (visited on 10/16/2020).
- Balakrishnan, Vimala et al. (Apr. 2019). "Cyberbullying detection on twitter using Big Five and Dark Triad features". en. In: *Personality and Individual Differences* 141, pp. 252–257. ISSN: 0191-8869. DOI: 10.1016/j.paid.2019.01.024. URL: <http://www.sciencedirect.com/science/article/pii/S0191886919300364> (visited on 05/06/2020).
- Banerjee, Priyanka et al. (2018). "Prediction Is a Balancing Act: Importance of Sampling Methods to Balance Sensitivity and Specificity of Predictive Models Based on Imbalanced Chemical Data Sets". English. In: *Frontiers in Chemistry* 6. ISSN:

- 2296-2646. DOI: 10.3389/fchem.2018.00362. URL: <https://www.frontiersin.org/articles/10.3389/fchem.2018.00362/full> (visited on 04/01/2021).
- Baroni, M. et al. (2014). "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". In: vol. 1, pp. 238–247. DOI: 10.3115/v1/p14-1023.
- Barry, James (2017). "Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches". en. In: *AICS*, pp. 272–274.
- Basgall, María José et al. (2019). "An Analysis of Local and Global Solutions to Address Big Data Imbalanced Classification: A Case Study with SMOTE Preprocessing". en. In: *Cloud Computing and Big Data*. Ed. by Marcelo Naiouf et al. Communications in Computer and Information Science. Cham: Springer International Publishing, pp. 75–85. ISBN: 978-3-030-27713-0. DOI: 10.1007/978-3-030-27713-0\_7.
- Batista, Gustavo E. A. P. A. et al. (June 2004). "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD Explorations Newsletter* 6.1, pp. 20–29. ISSN: 1931-0145. DOI: 10.1145/1007730.1007735. URL: <https://dl.acm.org/doi/10.1145/1007730.1007735> (visited on 10/13/2023).
- Bauder, Richard A. et al. (Sept. 2018). "The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data". en. In: *Health Information Science and Systems* 6.1, p. 9. ISSN: 2047-2501. DOI: 10.1007/s13755-018-0051-3. URL: <https://doi.org/10.1007/s13755-018-0051-3> (visited on 08/10/2020).
- Bergstra, James et al. (Feb. 2012). "Random search for hyper-parameter optimization". In: *J. Mach. Learn. Res.* 13.null, pp. 281–305. ISSN: 1532-4435. URL: <https://dl.acm.org/doi/10.5555/2188385.2188395> (visited on 03/14/2026).
- Bollen, Johan et al. (Mar. 2011). "Twitter mood predicts the stock market". In: *Journal of Computational Science* 2.1, pp. 1–8. ISSN: 18777503. DOI: 10.1016/j.jocs.2010.12.007. URL: <http://arxiv.org/abs/1010.3003> (visited on 05/13/2020).

- Bradley, Andrew P. (July 1997). "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern Recognition* 30.7, pp. 1145–1159. ISSN: 0031-3203. DOI: 10.1016/S0031-3203(96)00142-2. URL: [https://doi.org/10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2) (visited on 09/01/2023).
- Bravo-Marquez, Felipe et al. (2019). "AffectiveTweets: a Weka Package for Analyzing Affect in Tweets". In: *Journal of Machine Learning Research* 20.92, pp. 1–6. ISSN: 1533-7928. URL: <http://jmlr.org/papers/v20/18-450.html> (visited on 12/05/2024).
- Breiman, Leo (Oct. 2001). "Random Forests". en. In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: <https://doi.org/10.1023/A:1010933404324> (visited on 08/23/2023).
- Butnaru, Andrei M. et al. (Jan. 2017). "From Image to Text Classification: A Novel Approach based on Clustering Word Embeddings". en. In: *Procedia Computer Science. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France* 112, pp. 1783–1792. ISSN: 1877-0509. DOI: 10.1016/j.procs.2017.08.211. URL: <https://www.sciencedirect.com/science/article/pii/S1877050917316071> (visited on 04/21/2021).
- Byod, Ryan (2013). *RIOT Scan | Recursive Inspection of Text*. en-US. URL: <https://www.ryanboyd.io/software/riot/> (visited on 08/15/2023).
- Carducci, Giulio et al. (May 2018). "TwitPersonality: Computing Personality Traits from Tweets Using Word Embeddings and Supervised Learning". en. In: *Information* 9.5, p. 127. DOI: 10.3390/info9050127. URL: <https://www.mdpi.com/2078-2489/9/5/127> (visited on 05/07/2020).
- Caropreso, Maria Fernanda et al. (2006). "Beyond the Bag of Words: A Text Representation for Sentence Selection". en. In: *Advances in Artificial Intelligence*. Ed. by Luc Lamontagne et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 324–335. ISBN: 978-3-540-34630-2. DOI: 10.1007/11766247\_28.

- Carrillo-de-Albornoz, Jorge et al. (Nov. 2018). "Feature engineering for sentiment analysis in e-health forums". en. In: *PLOS ONE* 13.11. Ed. by Nicholas D. Duran, e0207996. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0207996. URL: <https://dx.plos.org/10.1371/journal.pone.0207996> (visited on 06/05/2021).
- Cawley, Gavin C. et al. (Aug. 2010). "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation". In: *J. Mach. Learn. Res.* 11, pp. 2079–2107. ISSN: 1532-4435. URL: <https://dl.acm.org/doi/10.5555/1756006.1859921> (visited on 03/14/2026).
- Chatzakou, Despoina, Nicolas Kourtellis, et al. (June 2017). "Mean Birds: Detecting Aggression and Bullying on Twitter". In: *Proceedings of the 2017 ACM on Web Science Conference*. WebSci '17. Troy, New York, USA: Association for Computing Machinery, pp. 13–22. ISBN: 978-1-4503-4896-6. DOI: 10.1145/3091478.3091487. URL: <https://doi.org/10.1145/3091478.3091487> (visited on 05/05/2020).
- Chatzakou, Despoina, Athena Vakali, et al. (Dec. 2017). "Detecting variation of emotions in online activities". en. In: *Expert Systems with Applications* 89, pp. 318–332. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2017.07.044. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417305213> (visited on 11/11/2020).
- Chawla, N. V. et al. (June 2002). "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16, pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953. URL: <http://arxiv.org/abs/1106.1813> (visited on 08/26/2019).
- Chen, Chao et al. (2004). *Using Random Forest to Learn Imbalanced Data* | Department of Statistics. Tech. rep. 666. University of California, Berkeley, pp. 1–12. URL: <https://statistics.berkeley.edu/tech-reports/666> (visited on 11/16/2020).
- Chen, Enhong et al. (Mar. 2011). "Exploiting probabilistic topic models to improve text categorization under class imbalance". en. In: *Information Processing & Management* 47.2, pp. 202–214. ISSN: 03064573. DOI: 10.1016/j.ipm.2010.07.003.

- URL: <https://linkinghub.elsevier.com/retrieve/pii/S0306457310000622> (visited on 08/17/2020).
- Chen, Wuxing et al. (May 2024). "A survey on imbalanced learning: latest research, applications and future directions". en. In: *Artificial Intelligence Review* 57.6, p. 137. ISSN: 1573-7462. DOI: 10.1007/s10462-024-10759-6. URL: <https://doi.org/10.1007/s10462-024-10759-6> (visited on 12/09/2024).
- Chen, Yu et al. (2021). "Effects of Data Augmentation Method Borderline-SMOTE on Emotion Recognition of EEG Signals Based on Convolutional Neural Network". In: *IEEE Access* 9. Conference Name: IEEE Access, pp. 47491–47502. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3068316. URL: <https://ieeexplore.ieee.org/document/9385376/?arnumber=9385376> (visited on 07/16/2024).
- Cheng, Tao et al. (June 2014). "Event Detection using Twitter: A Spatio-Temporal Approach". In: *PLoS ONE* 9.6, e97807. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0097807. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4043742/> (visited on 11/29/2023).
- Chung, Cindy et al. (2007). "The Psychological Functions of Function Words". In: *Social communication*. Frontiers of social psychology. New York, NY, US: Psychology Press, pp. 343–359. ISBN: 978-1-84169-428-3.
- Cichosz, P. (2018). "A case study in text mining of discussion forum posts: Classification with bag of words and global vectors". EN. In: *International Journal of Applied Mathematics and Computer Science* Vol. 28.no. 4. ISSN: 1641-876X. DOI: 10.2478/amcs-2018-0060. URL: <http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-d20e2cbb-0346-4a8c-9185-f4318377f9fe> (visited on 04/20/2021).
- Clark, Eleanor et al. (Jan. 2011). "Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English". en. In: *Procedia - Social and Behavioral Sciences*. Computational Linguistics and Related Fields 27, pp. 2–11. ISSN: 1877-0428. DOI: 10.1016/j.sbspro.2011.10.577. URL:

<https://www.sciencedirect.com/science/article/pii/S1877042811024049>  
(visited on 05/24/2021).

Cohen, Jacob (Apr. 1960). "A Coefficient of Agreement for Nominal Scales". en. In: *Educational and Psychological Measurement* 20.1, pp. 37–46. ISSN: 0013-1644, 1552-3888. DOI: 10.1177/001316446002000104. URL: <http://journals.sagepub.com/doi/10.1177/001316446002000104> (visited on 09/07/2023).

Cover, T. et al. (Jan. 1967). "Nearest neighbor pattern classification". In: *IEEE Transactions on Information Theory* 13.1. Conference Name: IEEE Transactions on Information Theory, pp. 21–27. ISSN: 1557-9654. DOI: 10.1109/TIT.1967.1053964.

Dadvar, M. et al. (Feb. 2012). "Improved cyberbullying detection using gender information". Undefined. In: *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*. Ghent University, pp. 23–25. URL: <https://research.utwente.nl/en/publications/improved-cyberbullying-detection-using-gender-information> (visited on 08/01/2019).

Dadvar, Maral et al. (2012). "Cyberbullying detection: a step toward a safer internet yard". en. In: *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. Lyon, France: ACM Press, p. 121. ISBN: 978-1-4503-1230-1. DOI: 10.1145/2187980.2187995. URL: <http://dl.acm.org/citation.cfm?doid=2187980.2187995> (visited on 08/11/2019).

Dai, Andrew M. et al. (Nov. 2015). "Semi-supervised Sequence Learning". In: *arXiv:1511.01432 [cs]*. URL: <http://arxiv.org/abs/1511.01432> (visited on 04/19/2021).

Damaschk, Matthias et al. (Sept. 2019). "Multiclass Text Classification on Unbalanced, Sparse and Noisy Data". In: *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*. Turku, Finland: Linköping University Electronic Press, pp. 58–65. URL: <https://www.aclweb.org/anthology/W19-6207> (visited on 08/17/2020).

Dang, Nhan Cach et al. (Mar. 2020). "Sentiment Analysis Based on Deep Learning: A Comparative Study". In: *Electronics* 9.3, p. 483. ISSN: 2079-9292. DOI: 10.3390/

- electronics9030483. URL: <http://arxiv.org/abs/2006.03541> (visited on 04/21/2021).
- Devlin, Jacob et al. (May 2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs]. DOI: 10.48550/arXiv.1810.04805. URL: <http://arxiv.org/abs/1810.04805> (visited on 07/17/2024).
- Dilmaç, B. (2009). "Psychological needs as a predictor of cyber bullying: A preliminary report on college students". In: *Kuram ve Uygulamada Eğitim Bilimleri* 9.3, pp. 1307–1325.
- Drummond, Chris et al. (2000). "Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria". en. In: *Proceedings of Seventeenth International Conference on Machine Learning*, pp. 239–246.
- Du, Changshun et al. (Feb. 2018). "Text Classification Research with Attention-based Recurrent Neural Networks". en. In: *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL* 13.1. Number: 1, pp. 50–61. ISSN: 1841-9844. URL: <https://univagora.ro/jour/index.php/ijccc/article/view/3142> (visited on 07/14/2024).
- Du, Li-min et al. (Sept. 2015). "Feature Selection for Multi-Class Imbalanced Data Sets Based on Genetic Algorithm". en. In: *Annals of Data Science* 2.3, pp. 293–300. ISSN: 2198-5812. DOI: 10.1007/s40745-015-0060-x. URL: <https://doi.org/10.1007/s40745-015-0060-x> (visited on 03/24/2021).
- Du, Xu et al. (2020). "An Integrated Framework Based on Latent Variational Autoencoder for Providing Early Warning of At-Risk Students". In: *IEEE Access* 8. Conference Name: IEEE Access, pp. 10110–10122. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2964845. URL: <https://ieeexplore.ieee.org/document/8952699/?arnumber=8952699> (visited on 07/16/2024).
- Duangsoithong, Rakkrit et al. (Apr. 2010). "Correlation-Based and Causal Feature Selection Analysis for Ensemble Classifiers". In: *Artificial Neural Networks in Pattern Recognition, 4th IAPR TC3 Workshop, ANNPR 2010, Proceedings*. Cairo, Egypt, pp. 25–36. ISBN: 978-3-642-12158-6. DOI: 10.1007/978-3-642-12159-3\_3.

- Duggan, Maeve (Oct. 2014). *Online Harassment*. en-US. URL: <https://www.pewinternet.org/2014/10/22/online-harassment/> (visited on 08/17/2019).
- Duong, Dat et al. (Jan. 2017). "A novel Word2vec based tool to estimate semantic similarity of genes by using Gene Ontology terms". en. In: *bioRxiv*, p. 103648. DOI: 10.1101/103648. URL: <https://www.biorxiv.org/content/10.1101/103648v1> (visited on 04/22/2021).
- Duwairi, Rehab et al. (Aug. 2014). "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text". In: *Journal of Information Science* 40.4, pp. 501–513. ISSN: 0165-5515. DOI: 10.1177/0165551514534143. URL: <https://doi.org/10.1177/0165551514534143> (visited on 05/27/2021).
- Eichstaedt, Johannes C. et al. (Feb. 2015). "Psychological Language on Twitter Predicts County-Level Heart Disease Mortality". en. In: *Psychological Science* 26.2, pp. 159–169. ISSN: 0956-7976, 1467-9280. DOI: 10.1177/0956797614557867. URL: <http://journals.sagepub.com/doi/10.1177/0956797614557867> (visited on 05/13/2020).
- Einarsen, Stale et al. (Oct. 2002). *Bullying and Emotional Abuse in the Workplace: International Perspectives in Research and Practice*. en. CRC Press. ISBN: 978-0-203-16466-2.
- Eklund, Martin (2018). "Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data". en. PhD thesis. Stockholm, Sweden: Kth Royal Institute Of Technology.
- Elkan, Charles (2001). "The Foundations of Cost-Sensitive Learning". en. In: *In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence*, 973–978. Seattle, Washington: Morgan Kaufmann, p. 6.
- Elrahman, Shaza M Abd et al. (2013). "A Review of Class Imbalance Problem". en. In: 1.2013, pp. 332–340. ISSN: 2160-2174.
- Esmeli, Ramazan et al. (July 2020). "Using Word2Vec Recommendation for Improved Purchase Prediction". In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9206871.

- Fan, Zhengyu et al. (Nov. 2017). "Multi-label Chinese question classification based on word2vec". In: *2017 4th International Conference on Systems and Informatics (ICSAI)*, pp. 546–550. DOI: 10.1109/ICSAI.2017.8248351.
- Fawcett, Tom (June 2006). "An introduction to ROC analysis". In: *Pattern Recognition Letters*. ROC Analysis in Pattern Recognition 27.8, pp. 861–874. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2005.10.010. URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X> (visited on 08/15/2023).
- Feng, Mengke et al. (Mar. 2016). "A Distributed Chinese Naive Bayes Classifier Based on Word Embedding". en. In: Atlantis Press, pp. 1120–1126. ISBN: 978-94-6252-165-0. DOI: 10.2991/icmmct-16.2016.222. URL: <https://www.atlantispress.com/proceedings/icmmct-16/25850121> (visited on 04/22/2021).
- Fernandez, Alberto et al. (Apr. 2018). "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary". en. In: *Journal of Artificial Intelligence Research* 61, pp. 863–905. ISSN: 1076-9757. DOI: 10.1613/jair.1.11192. URL: <https://www.jair.org/index.php/jair/article/view/11192> (visited on 03/27/2021).
- Festl, Ruth et al. (Jan. 2013). "Social Relations and Cyberbullying: the Influence of Individual and Structural Attributes on Victimization and Perpetration via the Internet". en. In: *Human Communication Research* 39.1, pp. 101–126. ISSN: 0360-3989. DOI: 10.1111/j.1468-2958.2012.01442.x. URL: <https://academic.oup.com/hcr/article/39/1/101/4093695> (visited on 05/08/2020).
- Feurer, Matthias et al. (2019). "Auto-sklearn: Efficient and Robust Automated Machine Learning". en. In: *Automated Machine Learning: Methods, Systems, Challenges*. Ed. by Frank Hutter et al. Cham: Springer International Publishing, pp. 113–134. ISBN: 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5\_6. URL: [https://doi.org/10.1007/978-3-030-05318-5\\_6](https://doi.org/10.1007/978-3-030-05318-5_6) (visited on 07/17/2024).
- Finn, Chelsea et al. (July 2017). *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. arXiv:1703.03400 [cs]. DOI: 10.48550/arXiv.1703.03400. URL: <http://arxiv.org/abs/1703.03400> (visited on 07/17/2024).

- Forman, George (Mar. 2003). "An extensive empirical study of feature selection metrics for text classification". In: *The Journal of Machine Learning Research* 3.null, pp. 1289–1305. ISSN: 1532-4435.
- Fürnkranz, Johannes et al. (Nov. 2008). "Multilabel classification via calibrated label ranking". en. In: *Machine Learning* 73.2, pp. 133–153. ISSN: 1573-0565. DOI: 10.1007/s10994-008-5064-8. URL: <https://doi.org/10.1007/s10994-008-5064-8> (visited on 07/14/2024).
- Galán-García, Patxi et al. (2014). "Supervised Machine Learning for the Detection of Troll Profiles in Twitter Social Network: Application to a Real Case of Cyberbullying". en. In: *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*. Ed. by Álvaro Herrero et al. Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 419–428. ISBN: 978-3-319-01854-6.
- Gao, Kehan et al. (July 2015). "Combining Feature Subset Selection and Data Sampling for Coping with Highly Imbalanced Software Data". en. In: pp. 439–444. DOI: 10.18293/SEKE2015-182. URL: [http://ksiresearchorg.ipage.com/seke/seke15paper/seke15paper\\_182.pdf](http://ksiresearchorg.ipage.com/seke/seke15paper/seke15paper_182.pdf) (visited on 10/03/2023).
- Al-garadi, Mohammed Ali et al. (Oct. 2016). "Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network". en. In: *Computers in Human Behavior* 63, pp. 433–443. ISSN: 0747-5632. DOI: 10.1016/j.chb.2016.05.051. URL: <http://www.sciencedirect.com/science/article/pii/S0747563216303788> (visited on 05/07/2020).
- Garrett, Michael et al. (2018). "Leveraging Mutual Information to Generate Domain Specific Lexicons". en. In: *International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction and Behavior Representation in Modeling and Simulation*. Washington DC, USA, pp. 1–7.
- Ge, Lihao et al. (Dec. 2017). "Improving text classification with word embedding". In: *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1796–1805. DOI: 10.1109/BigData.2017.8258123.

- Geel, Mitch van et al. (Feb. 2017). "Which personality traits are related to traditional bullying and cyberbullying? A study with the Big Five, Dark Triad and sadism". en. In: *Personality and Individual Differences* 106, pp. 231–235. ISSN: 01918869. DOI: 10.1016/j.paid.2016.10.063. URL: <https://linkinghub.elsevier.com/retrieve/pii/S019188691631100X> (visited on 05/08/2020).
- Ghag, Kranti Vithal et al. (Sept. 2015). "Comparative analysis of effect of stopwords removal on sentiment classification". In: *2015 International Conference on Computer, Communication and Control (IC4)*, pp. 1–6. DOI: 10.1109/IC4.2015.7375527.
- Ghosh, Kushankur et al. (Oct. 2019). "Imbalanced Twitter Sentiment Analysis using Minority Oversampling". In: *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pp. 1–5. DOI: 10.1109/ICAwST.2019.8923218.
- Gimpel, Kevin et al. (June 2011). "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 42–47. URL: <https://www.aclweb.org/anthology/P11-2008> (visited on 09/10/2019).
- Glazkova, Anna (Aug. 2020). "A Comparison of Synthetic Oversampling Methods for Multi-class Text Classification". In: *arXiv:2008.04636 [cs]*. URL: <http://arxiv.org/abs/2008.04636> (visited on 04/23/2021).
- Goncalves, Carlos Adriano et al. (2010). "The Impact of Pre-processing on the Classification of MEDLINE Documents." en. In: *Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems*. Funchal, Madeira, Portugal: SciTePress - Science, pp. 53–61. ISBN: 978-989-8425-14-0. DOI: 10.5220/0003028700530061. URL: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0003028700530061> (visited on 04/28/2021).
- González-Bailón, Sandra et al. (July 2014). "Assessing the bias in samples of large online networks". en. In: *Social Networks* 38, pp. 16–27. ISSN: 03788733. DOI: 10.1016/j.socnet.2014.01.004. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0378873314000057> (visited on 06/12/2019).

- Graczyk, Magdalena et al. (2010). "Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal". en. In: *Intelligent Information and Database Systems*. Ed. by Ngoc Thanh Nguyen et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 340–350. ISBN: 978-3-642-12101-2. DOI: 10.1007/978-3-642-12101-2\_35.
- Grootendorst, Maarten et al. (2020). "Beyond Bag-of-Concepts: Vectors of Locally Aggregated Concepts". en. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Ulf Brefeld et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 681–696. ISBN: 978-3-030-46147-8. DOI: 10.1007/978-3-030-46147-8\_41.
- Guyon, Isabelle et al. (2003). "An Introduction to Variable and Feature Selection". en. In: *Journal of Machine Learning Research* 3, pp. 1157–1182.
- HaCohen-Kerner, Yaakov, Daniel Miller, et al. (May 2020). "The influence of pre-processing on text classification using a bag-of-words representation". In: *PLoS ONE* 15.5, e0232525. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0232525. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7194364/> (visited on 10/29/2023).
- HaCohen-Kerner, Yaakov, Avi Rosenfeld, et al. (Jan. 2018). "Topic-based Classification through Unigram Unmasking". In: *Procedia Computer Science*. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia 126, pp. 69–76. ISSN: 1877-0509. DOI: 10.1016/j.procs.2018.07.210. URL: <https://www.sciencedirect.com/science/article/pii/S1877050918311840> (visited on 08/24/2023).
- Haddi, Emma et al. (2013). "The Role of Text Pre-processing in Sentiment Analysis". en. In: *Procedia Computer Science* 17, pp. 26–32. ISSN: 18770509. DOI: 10.1016/j.procs.2013.05.005. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1877050913001385> (visited on 11/15/2020).
- Haixiang, Guo et al. (May 2017). "Learning from class-imbalanced data: Review of methods and applications". en. In: *Expert Systems with Applications* 73, pp. 220–

239. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2016.12.035. URL: <http://www.sciencedirect.com/science/article/pii/S0957417416307175> (visited on 06/18/2020).
- Hall, Mark A. (May 2000). *Correlation-based feature selection of discrete and numeric class machine learning*. en. Working Paper. Accepted: 2008-10-13T03:42:21Z ISSN: 1170-487X. University of Waikato, Department of Computer Science. URL: <https://researchcommons.waikato.ac.nz/handle/10289/1024> (visited on 10/01/2023).
- Hand, David J. et al. (Nov. 2001). "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". en. In: *Machine Learning* 45.2, pp. 171–186. ISSN: 1573-0565. DOI: 10.1023/A:1010920819831. URL: <https://doi.org/10.1023/A:1010920819831> (visited on 04/14/2021).
- Al-Harbi, Omar (Feb. 2019). "A Comparative Study of Feature Selection Methods for Dialectal Arabic Sentiment Classification Using Support Vector Machine". In: 19.1. arXiv:1902.06242 [cs], pp. 167–176. URL: <http://arxiv.org/abs/1902.06242> (visited on 10/02/2023).
- Hasan, Basna Mohammed Salih et al. (Apr. 2021). "A Review of Principal Component Analysis Algorithm for Dimensionality Reduction". en. In: *Journal of Soft Computing and Data Mining* 2.1. Number: 1, pp. 20–30. ISSN: 2716-621X. URL: <https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/8032> (visited on 12/24/2024).
- Hasan, Sakib Al et al. (Feb. 2022). "Classification of Multi-Labeled Text Articles with Reuters Dataset using SVM". In: *2022 International Conference on Science and Technology (ICOSTECH)*, pp. 01–05. DOI: 10.1109/ICOSTECH54296.2022.9829153. URL: <https://ieeexplore.ieee.org/abstract/document/9829153> (visited on 07/14/2024).
- Hasanin, Tawfiq et al. (Nov. 2019). "Severely imbalanced Big Data challenges: investigating data sampling approaches". In: *Journal of Big Data* 6.1, p. 107. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0274-4. URL: <https://doi.org/10.1186/s40537-019-0274-4> (visited on 08/18/2023).

- He, H. et al. (Sept. 2009). "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9, pp. 1263–1284. ISSN: 1041-4347. DOI: 10.1109/TKDE.2008.239.
- Heap, Bradford et al. (Sept. 2017). "Word Vector Enrichment of Low Frequency Words in the Bag-of-Words Model for Short Text Multi-class Classification Problems". In: *arXiv:1709.05778 [cs]*. URL: <http://arxiv.org/abs/1709.05778> (visited on 05/31/2021).
- Holtgraves, Thomas (Feb. 2011). "Text messaging, personality, and the social context". en. In: *Journal of Research in Personality* 45.1, pp. 92–99. ISSN: 0092-6566. DOI: 10.1016/j.jrp.2010.11.015. URL: <http://www.sciencedirect.com/science/article/pii/S0092656610001698> (visited on 05/08/2020).
- Hoover, Kristine (Jan. 2015). "Reflections on Student Affect in Group Projects: Are We Encouraging Our Students to Be in Community with Others?" In: *Jesuit Higher Education: A Journal* 4.1. ISSN: 2164-7666. DOI: -. URL: <https://epublications.regis.edu/jhe/vol4/iss1/11>.
- Hosseini, Monireh et al. (Mar. 2016). "Recognizing users gender in social media using linguistic features". en. In: *Computers in Human Behavior* 56, pp. 192–197. ISSN: 0747-5632. DOI: 10.1016/j.chb.2015.11.049. URL: <http://www.sciencedirect.com/science/article/pii/S0747563215302648> (visited on 05/09/2020).
- Huang, Qianjia et al. (2014). "Cyber Bullying Detection Using Social and Textual Analysis". In: *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*. SAM '14. New York, NY, USA: ACM, pp. 3–6. ISBN: 978-1-4503-3124-1. DOI: 10.1145/2661126.2661133. URL: <http://doi.acm.org/10.1145/2661126.2661133> (visited on 06/26/2019).
- Imran, Muhammad et al. (2016). "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages". en. In: p. 6.
- Iranmehr, Arya et al. (May 2019). "Cost-sensitive support vector machines". en. In: *Neurocomputing*. Learning in the Presence of Class Imbalance and Concept Drift 343, pp. 50–64. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2018.11.099. URL:

- <https://www.sciencedirect.com/science/article/pii/S0925231219301614> (visited on 03/24/2021).
- Jain, Anil et al. (Dec. 2005). "Score normalization in multimodal biometric systems". In: *Pattern Recognition* 38.12, pp. 2270–2285. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2005.01.012. URL: <https://www.sciencedirect.com/science/article/pii/S0031320305000592> (visited on 12/24/2024).
- Japkowicz, Nathalie (2000). "Learning from Imbalanced Data Sets: A Comparison of Various Strategies". en. In: vol. 68, pp. 10–15.
- (2008). "Classifier Evaluation: A Need for Better Education and Restructuring". en. In: *In Proceedings of the 3rd Workshop on Evaluation Methods for Machine Learning, ICML*, pp. 1–4.
- Japkowicz, Nathalie and Shaju Stephen (Oct. 2002). "The class imbalance problem: A systematic study". In: *Intelligent Data Analysis* 6.5, pp. 429–449. ISSN: 1088-467X.
- Java, Akshay et al. (2007). "Why we twitter: understanding microblogging usage and communities". en. In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD '07*. San Jose, California: ACM Press, pp. 56–65. ISBN: 978-1-59593-848-0. DOI: 10.1145/1348549.1348556. URL: <http://portal.acm.org/citation.cfm?doid=1348549.1348556> (visited on 05/13/2020).
- Jiang, Shengyi et al. (Jan. 2012). "An improved K-nearest-neighbor algorithm for text categorization". en. In: *Expert Systems with Applications* 39.1, pp. 1503–1509. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2011.08.040. URL: <https://www.sciencedirect.com/science/article/pii/S0957417411011511> (visited on 03/14/2021).
- Jiang, Suqi et al. (Apr. 2016). "Integrating rich document representations for text classification". In: *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, pp. 303–308. DOI: 10.1109/SIEDS.2016.7489319.

- Jianqiang, Zhao (Dec. 2015). "Pre-processing Boosting Twitter Sentiment Analysis?" In: *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 748–753. DOI: 10.1109/SmartCity.2015.158.
- Jianqiang, Zhao and Gui Xiaolin (2017). "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis". In: *IEEE Access* 5, pp. 2870–2879. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2672677.
- Joachims, Thorsten (1998). "Text categorization with Support Vector Machines: Learning with many relevant features". en. In: *Machine Learning: ECML-98*. Ed. by Jaime G. Carbonell et al. Vol. 1398. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 137–142. ISBN: 978-3-540-64417-0 978-3-540-69781-7. DOI: 10.1007/BFb0026683. URL: <http://link.springer.com/10.1007/BFb0026683> (visited on 08/22/2023).
- Johnson, Justin M. et al. (Dec. 2019). "Survey on deep learning with class imbalance". en. In: *Journal of Big Data* 6.1, p. 27. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0192-5. URL: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0192-5> (visited on 08/17/2020).
- Joshi, M. V. et al. (Nov. 2001). "Evaluating boosting algorithms to classify rare classes: comparison and improvements". In: *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 257–264. DOI: 10.1109/ICDM.2001.989527.
- Kamal, Abu H. M. et al. (Mar. 2010). "Feature selection for datasets with imbalanced class distributions". In: *International Journal of Software Engineering and Knowledge Engineering* 20.02, pp. 113–137. ISSN: 0218-1940. DOI: 10.1142/S0218194010004645. URL: <https://www.worldscientific.com/doi/10.1142/S0218194010004645> (visited on 04/04/2021).
- Kanan, Tarek et al. (2016). "Automated arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy". en. In: *Journal of the Association for Information Science and Technology* 67.11, pp. 2667–2683. ISSN: 2330-1643. DOI: 10.1002/asi.23609. URL: <https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.23609> (visited on 05/24/2021).

- Kern, Margaret L. et al. (Dec. 2013). "The Online Social Self: An Open Vocabulary Approach to Personality". en. In: *Assessment*. DOI: 10.1177/1073191113514104. URL: <https://journals.sagepub.com/doi/10.1177/1073191113514104> (visited on 05/08/2020).
- Khan, Salman H. et al. (Aug. 2017). "Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data". In: *arXiv:1508.03422 [cs]* 29.8, pp. 3573–3587. DOI: 10.1109/TNNLS.2017.2732482. URL: <http://arxiv.org/abs/1508.03422> (visited on 03/24/2021).
- Khoshgoftaar, T. M. et al. (Oct. 2007). "An Empirical Study of Learning from Imbalanced Data Using Random Forest". In: *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*. Vol. 2, pp. 310–317. DOI: 10.1109/ICTAI.2007.46.
- Khoshgoftaar, Taghi M. et al. (Sept. 2004). "Comparative Assessment of Software Quality Classification Techniques: An Empirical Case Study". en. In: *Empirical Software Engineering* 9.3, pp. 229–257. ISSN: 1573-7616. DOI: 10.1023/B:EMSE.0000027781.18360.9b. URL: <https://doi.org/10.1023/B:EMSE.0000027781.18360.9b> (visited on 09/01/2023).
- Kim, Jihye et al. (2021). "Mirroring Vector Space Embedding for New Words". In: *IEEE Access* 9. Conference Name: IEEE Access, pp. 99954–99967. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3096238. URL: <https://ieeexplore.ieee.org/abstract/document/9481109> (visited on 12/24/2024).
- Kowalski, Robin M. et al. (Jan. 2012). *Chapter 14 Cyber Bullying Among College Students: Evidence from Multiple Domains of College Life*. en. book-part. URL: [https://www.emerald.com/insight/content/doi/10.1108/S2044-9968\(2012\)0000005016/full/html](https://www.emerald.com/insight/content/doi/10.1108/S2044-9968(2012)0000005016/full/html) (visited on 09/04/2019).
- Kowsari, Kamran et al. (Apr. 2019). "Text Classification Algorithms: A Survey". In: *Information* 10.4. arXiv:1904.08067 [cs, stat], p. 150. ISSN: 2078-2489. DOI: 10.3390/info10040150. URL: <http://arxiv.org/abs/1904.08067> (visited on 08/18/2023).

- Krawczyk, Bartosz (Nov. 2016). "Learning from imbalanced data: open challenges and future directions". en. In: *Progress in Artificial Intelligence* 5.4, pp. 221–232. ISSN: 2192-6360. DOI: 10.1007/s13748-016-0094-0. URL: <https://doi.org/10.1007/s13748-016-0094-0> (visited on 04/04/2021).
- Krouska, Akrivi et al. (July 2016). "The effect of preprocessing techniques on Twitter sentiment analysis". In: *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, pp. 1–5. DOI: 10.1109/IISA.2016.7785373.
- Kubat, Miroslav, Robert C. Holte, et al. (Feb. 1998). "Machine Learning for the Detection of Oil Spills in Satellite Radar Images". en. In: *Machine Learning* 30.2, pp. 195–215. ISSN: 1573-0565. DOI: 10.1023/A:1007452223027. URL: <https://doi.org/10.1023/A:1007452223027> (visited on 09/01/2023).
- Kubat, Miroslav and Stan Matwin (Jan. 1997). "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection." en. In: *Proceedings of the 14th International Conference on Machine Learning*, pp. 179–186. URL: <https://www.scinapse.io/papers/85350352> (visited on 09/01/2023).
- Kucharczyk, Łukasz et al. (2014). "Evaluation of Path Based Methods for Conceptual Representation of the Text". en. In: *Foundations of Intelligent Systems*. Ed. by Troels Andreasen et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 435–444. ISBN: 978-3-319-08326-1. DOI: 10.1007/978-3-319-08326-1\_44.
- Kwak, Haewoon et al. (2010). "What is Twitter, a social network or a news media?" en. In: *Proceedings of the 19th international conference on World wide web - WWW '10*. Raleigh, North Carolina, USA: ACM Press, p. 591. ISBN: 978-1-60558-799-8. DOI: 10.1145/1772690.1772751. URL: <http://portal.acm.org/citation.cfm?doid=1772690.1772751> (visited on 05/13/2020).
- Kwak, N. et al. (Jan. 2002). "Input feature selection for classification problems". In: *IEEE Transactions on Neural Networks* 13.1, pp. 143–159. ISSN: 1941-0093. DOI: 10.1109/72.977291.

- Landis, J. R. et al. (Mar. 1977). "The measurement of observer agreement for categorical data". eng. In: *Biometrics* 33.1, pp. 159–174. ISSN: 0006-341X.
- Lango, Mateusz (June 2019). "Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study". en. In: *Foundations of Computing and Decision Sciences* 44.2, pp. 151–178. DOI: 10.2478/fcds-2019-0009. URL: <https://content.sciendo.com/view/journals/fcds/44/2/article-p151.xml> (visited on 08/10/2020).
- Lango, Mateusz and Jerzy Stefanowski (Feb. 2018). "Multi-class and feature selection extensions of Roughly Balanced Bagging for imbalanced data". en. In: *Journal of Intelligent Information Systems* 50.1, pp. 97–127. ISSN: 1573-7675. DOI: 10.1007/s10844-017-0446-7. URL: <https://doi.org/10.1007/s10844-017-0446-7> (visited on 04/04/2021).
- Lee, Changki et al. (Jan. 2006). "Information gain and divergence-based feature selection for machine learning-based text categorization". In: *Information Processing & Management. Formal Methods for Information Retrieval* 42.1, pp. 155–165. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2004.08.006. URL: <https://www.sciencedirect.com/science/article/pii/S0306457304000962> (visited on 10/01/2023).
- Leevy, Joffrey L. et al. (Nov. 2018). "A survey on addressing high-class imbalance in big data". In: *Journal of Big Data* 5.1, p. 42. ISSN: 2196-1115. DOI: 10.1186/s40537-018-0151-6. URL: <https://doi.org/10.1186/s40537-018-0151-6> (visited on 08/10/2020).
- Li, Qian et al. (Dec. 2021). *A Survey on Text Classification: From Shallow to Deep Learning*. arXiv:2008.00364 [cs]. URL: <http://arxiv.org/abs/2008.00364> (visited on 08/22/2023).
- Li, Ruiguang et al. (2022). "A Review of Machine Learning Algorithms for Text Classification". en. In: *Cyber Security*. Ed. by Wei Lu et al. Communications in Computer and Information Science. Singapore: Springer Nature, pp. 226–234. ISBN: 9789811692291. DOI: 10.1007/978-981-16-9229-1\_14.

- Li, Yanling et al. (Oct. 2010). "Data Imbalance Problem in Text Classification". In: *2010 Third International Symposium on Information Processing*, pp. 301–305. DOI: 10.1109/ISIP.2010.47.
- Lilleberg, Joseph et al. (July 2015). "Support vector machines and Word2vec for text classification with semantic features". In: *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI\*CC)*, pp. 136–140. DOI: 10.1109/ICCI-CC.2015.7259377.
- Lin, W.-J. et al. (Jan. 2013). "Class-imbalanced classifiers for high-dimensional data". en. In: *Briefings in Bioinformatics* 14.1, pp. 13–26. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbs006. URL: <https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbs006> (visited on 12/24/2024).
- Lin, Ying-Dar et al. (2022). "Machine Learning With Variational AutoEncoder for Imbalanced Datasets in Intrusion Detection". In: *IEEE Access* 10. Conference Name: IEEE Access, pp. 15247–15260. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3149295. URL: <https://ieeexplore.ieee.org/document/9705580/?arnumber=9705580> (visited on 07/16/2024).
- Ling, Charles X et al. (2008). "Cost-Sensitive Learning and the Class Imbalance Problem". en. In: *in Encyclopedia of Machine Learning. C. Sammut (Ed.)*, p. 8. URL: [https://www.csd.uwo.ca/~xling/papers/cost\\_sensitive.pdf](https://www.csd.uwo.ca/~xling/papers/cost_sensitive.pdf).
- Ling, Charles X., Jin Huang, et al. (Aug. 2003). "AUC: a statistically consistent and more discriminating measure than accuracy". In: *Proceedings of the 18th international joint conference on Artificial intelligence. IJCAI'03*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 519–524. (Visited on 09/01/2023).
- Ling, Charles X. and Chenghui Li (Aug. 1998). "Data mining for direct marketing: problems and solutions". In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. KDD'98*. New York, NY: AAAI Press, pp. 73–79. (Visited on 09/22/2023).
- Ling, Charles X., Qiang Yang, et al. (2004). "Decision trees with minimal costs". en. In: *Twenty-first international conference on Machine learning - ICML '04*. Banff, Al-

- berta, Canada: ACM Press, p. 69. DOI: 10.1145/1015330.1015369. URL: <http://portal.acm.org/citation.cfm?doid=1015330.1015369> (visited on 03/27/2021).
- Liu, Jingzhou et al. (Aug. 2017). "Deep Learning for Extreme Multi-label Text Classification". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. New York, NY, USA: Association for Computing Machinery, pp. 115–124. ISBN: 978-1-4503-5022-8. DOI: 10.1145/3077136.3080834. URL: <https://doi.org/10.1145/3077136.3080834> (visited on 07/14/2024).
- Liu, Wendy et al. (2013). "What's in a Name? Using First Names as Features for Gender Inference in Twitter". en. In: *Analyzing Microtext: Papers from the 2013 AAAI Spring Symposium*. Stanford University, CA, USA, pp. 10–16.
- Liu, Ying et al. (Jan. 2009). "Imbalanced text classification: A term weighting approach". en. In: *Expert Systems with Applications* 36.1, pp. 690–701. ISSN: 09574174. DOI: 10.1016/j.eswa.2007.10.042. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0957417407005350> (visited on 06/05/2021).
- Liu, Xu-ying et al. (Dec. 2006). "The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study". en. In: *Sixth International Conference on Data Mining (ICDM'06)*. Hong Kong, China: IEEE, pp. 970–974. DOI: 10.1109/ICDM.2006.158. URL: <http://ieeexplore.ieee.org/document/4053137/> (visited on 08/26/2019).
- Liu, Zhiyue et al. (Apr. 2020). "CatGAN: Category-Aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05. Number: 05, pp. 8425–8432. ISSN: 2374-3468. DOI: 10.1609/aaai.v34i05.6361. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/6361> (visited on 07/16/2024).
- López, Victoria et al. (Nov. 2013). "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics". en. In: *Information Sciences* 250, pp. 113–141. ISSN: 0020-0255. DOI: 10.1016/j.

ins.2013.07.007. URL: <http://www.sciencedirect.com/science/article/pii/S0020025513005124> (visited on 11/14/2020).

Loyola-Gonzalez, Octavio et al. (2019). "Cost-Sensitive Pattern-Based classification for Class Imbalance problems". en. In: *IEEE Access* 7, pp. 60411–60427. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2913982. URL: <https://ieeexplore.ieee.org/document/8701690/> (visited on 11/28/2023).

Lozano, Aurélie C. et al. (Aug. 2008). "Multi-class cost-sensitive boosting with p-norm loss functions". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '08. New York, NY, USA: Association for Computing Machinery, pp. 506–514. ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401953. URL: <https://dl.acm.org/doi/10.1145/1401890.1401953> (visited on 09/01/2023).

Lu, Junru et al. (2016). "A Classification Method of Imbalanced Data Base on PSO Algorithm". en. In: *Social Computing*. Ed. by Wanxiang Che et al. Communications in Computer and Information Science. Singapore: Springer, pp. 121–134. ISBN: 978-981-10-2098-8. DOI: 10.1007/978-981-10-2098-8\_16.

Madabushi, Harish Tayyar et al. (Mar. 2020). "Cost-Sensitive BERT for Generalisable Sentence Classification with Imbalanced Data". In: *arXiv:2003.11563 [cs, stat]*. URL: <http://arxiv.org/abs/2003.11563> (visited on 08/17/2020).

Mahmud, Jalal et al. (2013). "Recommending targeted strangers from whom to solicit information on social media". en. In: *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*. Santa Monica, California, USA: ACM Press, p. 37. ISBN: 978-1-4503-1965-2. DOI: 10.1145/2449396.2449403. URL: <http://dl.acm.org/citation.cfm?doid=2449396.2449403> (visited on 05/07/2020).

Majumder, Anima et al. (Oct. 2020). "A Method for Handling Multi-class Imbalanced Data by Geometry based Information Sampling and Class Prioritized Synthetic Data Generation (GICaPS)". In: *arXiv:2010.05155 [cs, stat]*. URL: <http://arxiv.org/abs/2010.05155> (visited on 03/28/2021).

- Maldonado, Sebastián et al. (June 2018). “Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification”. In: *Applied Soft Computing* 67, pp. 94–105. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2018.02.051. URL: <https://www.sciencedirect.com/science/article/pii/S1568494618301108> (visited on 08/29/2023).
- Malhotra, Ruchika (Feb. 2015). “A systematic review of machine learning techniques for software fault prediction”. In: *Applied Soft Computing* 27, pp. 504–518. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2014.11.023. URL: <https://www.sciencedirect.com/science/article/pii/S1568494614005857> (visited on 08/24/2023).
- Mariani, Giovanni et al. (June 2018). *BAGAN: Data Augmentation with Balancing GAN*. arXiv:1803.09655 [cs, stat]. DOI: 10.48550/arXiv.1803.09655. URL: <http://arxiv.org/abs/1803.09655> (visited on 07/16/2024).
- Maron, M. E. (July 1961). “Automatic Indexing: An Experimental Inquiry”. en. In: *Journal of the ACM* 8.3, pp. 404–417. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/321075.321084. URL: <https://dl.acm.org/doi/10.1145/321075.321084> (visited on 08/22/2023).
- McHugh, Mary L. (2012). “Interrater reliability: the kappa statistic”. eng. In: *Biochemia Medica* 22.3, pp. 276–282. ISSN: 1330-0962.
- Mehta, Rajat (July 2017). *Big Data Analytics with Java*. en. Birmingham, United Kingdom: Packt Publishing Ltd. ISBN: 978-1-78728-219-3.
- Méndez, J. R. et al. (Oct. 2005). “Tokenising, stemming and stopword removal on anti-spam filtering domain”. In: *Proceedings of the 11th Spanish association conference on Current Topics in Artificial Intelligence*. CAEPIA’05. Berlin, Heidelberg: Springer-Verlag, pp. 449–458. ISBN: 978-3-540-45914-9. DOI: 10.1007/11881216\_47. URL: [https://doi.org/10.1007/11881216\\_47](https://doi.org/10.1007/11881216_47) (visited on 05/20/2021).
- Mikolov, Tomas, Kai Chen, et al. (Sept. 2013). “Efficient Estimation of Word Representations in Vector Space”. In: *arXiv:1301.3781 [cs]*. URL: <http://arxiv.org/abs/1301.3781> (visited on 11/14/2020).

- Mikolov, Tomas, Ilya Sutskever, et al. (Oct. 2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *arXiv:1310.4546 [cs, stat]*, pp. 3111–3119. URL: <http://arxiv.org/abs/1310.4546> (visited on 11/14/2020).
- Minaee, Shervin et al. (Apr. 2020). "Deep Learning Based Text Classification: A Comprehensive Review". In: *arXiv:2004.03705 [cs, stat]*. URL: <http://arxiv.org/abs/2004.03705> (visited on 09/17/2020).
- Mitsopoulou, Effrosyni et al. (Mar. 2015). "Personality traits, empathy and bullying behavior: A meta-analytic approach". en. In: *Aggression and Violent Behavior* 21, pp. 61–72. ISSN: 1359-1789. DOI: 10.1016/j.avb.2015.01.007. URL: <http://www.sciencedirect.com/science/article/pii/S1359178915000087> (visited on 05/09/2020).
- Mladenic, Dunja et al. (1999). "Feature selection for unbalanced class distribution and Naive Bayes". en. In: p. 10.
- Moayedikia, Alireza et al. (Jan. 2017). "Feature selection for high dimensional imbalanced class data using harmony search". In: *Engineering Applications of Artificial Intelligence* 57, pp. 38–49. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2016.10.008. URL: <https://www.sciencedirect.com/science/article/pii/S095219761630183X> (visited on 08/29/2023).
- Moraes, Rodrigo et al. (Mar. 2018). "Unbalanced sentiment classification: an assessment of ANN in the context of sampling the majority class". en. In: ISSN: 2167-9843, pp. 1–15. DOI: 10.7287/peerj.preprints.26618v1. URL: <https://peerj.com/preprints/26618> (visited on 09/14/2023).
- Moran, Sean et al. (July 2016). "Enhancing First Story Detection using Word Embeddings". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. SIGIR '16. New York, NY, USA: Association for Computing Machinery, pp. 821–824. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914719. URL: <https://doi.org/10.1145/2911451.2914719> (visited on 04/20/2021).

- Moreo, Alejandro et al. (2016). "Distributional Random Oversampling for Imbalanced Text Classification". en. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*. Pisa, Italy: ACM Press, pp. 805–808. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914722. URL: <http://dl.acm.org/citation.cfm?doid=2911451.2914722> (visited on 05/11/2020).
- Morstatter, Fred et al. (June 2013). "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose". In: *arXiv:1306.5204 [physics]*. URL: <http://arxiv.org/abs/1306.5204> (visited on 11/11/2020).
- Munková, Daša et al. (Jan. 2013). "Data Pre-processing Evaluation for Text Mining: Transaction/Sequence Model". en. In: *Procedia Computer Science*. 2013 International Conference on Computational Science 18, pp. 1198–1207. ISSN: 1877-0509. DOI: 10.1016/j.procs.2013.05.286. URL: <https://www.sciencedirect.com/science/article/pii/S1877050913004298> (visited on 05/24/2021).
- Napierala, Krystyna et al. (June 2015). "Types of minority class examples and their influence on learning classifiers from imbalanced data". en. In: *Journal of Intelligent Information Systems* 46.3, pp. 563–597. ISSN: 1573-7675. DOI: 10.1007/s10844-015-0368-1. URL: <https://doi.org/10.1007/s10844-015-0368-1> (visited on 03/23/2021).
- Navarro, Jordana N. et al. (Jan. 2012). "Going Cyber: Using Routine Activities Theory to Predict Cyberbullying Experiences". In: *Sociological Spectrum* 32.1, pp. 81–94. ISSN: 0273-2173. DOI: 10.1080/02732173.2012.628560. URL: <https://doi.org/10.1080/02732173.2012.628560> (visited on 10/14/2019).
- Nguyen, Heidi et al. (2018). "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches". en. In: 1.4, p. 23.
- O'Connor, Siobhan (2009). "A study of gender and age differences concerning the cyberbullying experiences of adolescents in Ireland". en. PhD thesis. Dublin, Ire-

- land: Dublin Business School. URL: <https://esource.dbs.ie/handle/10788/2377> (visited on 05/09/2020).
- Padurariu, Cristian et al. (2019). "Dealing with Data Imbalance in Text Classification". en. In: *Procedia Computer Science* 159, pp. 736–745. ISSN: 18770509. DOI: 10.1016/j.procs.2019.09.229. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1877050919314152> (visited on 08/17/2020).
- Palanivinayagam, Ashokkumar et al. (May 2023). "Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review". en. In: *Algorithms* 16.5. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 236. ISSN: 1999-4893. DOI: 10.3390/a16050236. URL: <https://www.mdpi.com/1999-4893/16/5/236> (visited on 12/03/2024).
- Parlak, Bekir (Sept. 2023). "Ensemble feature selection for single-label text classification: a comprehensive analytical study". en. In: *Neural Computing and Applications* 35.26, pp. 19235–19251. ISSN: 1433-3058. DOI: 10.1007/s00521-023-08763-y. URL: <https://doi.org/10.1007/s00521-023-08763-y> (visited on 12/24/2024).
- Pattnaik, Prasant Kumar et al. (Apr. 2018). *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2017*. en. Springer. ISBN: 978-981-10-7871-2.
- Pau, Simone et al. (Aug. 2023). "An Evaluation of Feature Selection Robustness on Class Noisy Data". en. In: *Information* 14.8. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 438. ISSN: 2078-2489. DOI: 10.3390/info14080438. URL: <https://www.mdpi.com/2078-2489/14/8/438> (visited on 10/02/2023).
- Pedregosa, Fabian et al. (June 2018). *Scikit-learn: Machine Learning in Python*. arXiv:1201.0490 [cs]. DOI: 10.48550/arXiv.1201.0490. URL: <http://arxiv.org/abs/1201.0490> (visited on 12/24/2024).
- Peersman, Claudia et al. (2011). "Predicting Age and Gender in Online Social Networks". In: *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*. SMUC '11. New York, NY, USA: ACM, pp. 37–44. ISBN: 978-

- 1-4503-0949-3. DOI: 10.1145/2065023.2065035. URL: <http://doi.acm.org/10.1145/2065023.2065035> (visited on 10/14/2019).
- Pennington, Jeffrey et al. (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: <https://aclanthology.org/D14-1162> (visited on 08/23/2023).
- Phoungphol, Piyaphol (Dec. 2013). "A Classification Framework for Imbalanced Data". en. PhD thesis. Georgia State University. DOI: 10.57709/4829222. URL: [https://scholarworks.gsu.edu/cs\\_diss/78](https://scholarworks.gsu.edu/cs_diss/78) (visited on 11/28/2023).
- Ah-Pine, Julien et al. (2016). "A Study of Synthetic Oversampling for Twitter Imbalanced Sentiment Analysis". en. In: p. 8.
- Pittaras, N. et al. (July 2021). "Text classification with semantically enriched word embeddings". en. In: *Natural Language Engineering* 27.4. Publisher: Cambridge University Press, pp. 391–425. ISSN: 1351-3249, 1469-8110. DOI: 10.1017/S1351324920000170. URL: <https://www.cambridge.org/core/journals/natural-language-engineering/article/text-classification-with-semantically-enriched-word-embeddings/44B2DCAED3F3058E09A35B01BF0DCF25> (visited on 08/28/2023).
- Pomikalek, Jan et al. (2007). "The Influence of preprocessing parameters on text categorization". In: *World Academy of Science, Engineering and Technology* 1, pp. 430–434.
- Potha, N. et al. (Dec. 2014). "Cyberbullying Detection using Time Series Modeling". In: *2014 IEEE International Conference on Data Mining Workshop*, pp. 373–382. DOI: 10.1109/ICDMW.2014.170.
- Preoțiu-Pietro, Daniel et al. (June 2015). "The role of personality, age, and gender in tweeting about mental illness". In: *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Denver, Colorado: Association for Computational Linguistics, pp. 21–30. DOI:

- 10.3115/v1/W15-1203. URL: <https://www.aclweb.org/anthology/W15-1203> (visited on 05/13/2020).
- Qiu, Jiangtao et al. (July 2018). “Leveraging sentiment analysis at the aspects level to predict ratings of reviews”. In: *Information Sciences* 451-452, pp. 295–309. ISSN: 0020-0255. DOI: 10.1016/j.ins.2018.04.009. URL: <https://www.sciencedirect.com/science/article/pii/S0020025516317303> (visited on 08/21/2023).
- Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. en. Morgan Kaufmann. ISBN: 978-1-55860-238-0.
- Ramos-López, Darío et al. (Jan. 2021). “Cost-Sensitive Variable Selection for Multi-Class Imbalanced Datasets Using Bayesian Networks”. en. In: *Mathematics* 9.2, p. 156. DOI: 10.3390/math9020156. URL: <https://www.mdpi.com/2227-7390/9/2/156> (visited on 03/24/2021).
- Rashidi, Hooman H. et al. (2023). “Common statistical concepts in the supervised Machine Learning arena”. In: *Frontiers in Oncology* 13. ISSN: 2234-943X. URL: <https://www.frontiersin.org/articles/10.3389/fonc.2023.1130229> (visited on 08/18/2023).
- Resett, Santiago et al. (Dec. 2017). “Traditional bullying and cyberbullying: Differences in emotional problems, and personality. Are cyberbullies more Machiavellians?” en. In: *Journal of Adolescence* 61, pp. 113–116. ISSN: 0140-1971. DOI: 10.1016/j.adolescence.2017.09.013. URL: <http://www.sciencedirect.com/science/article/pii/S0140197117301549> (visited on 05/08/2020).
- Reynolds, K. et al. (Dec. 2011). “Using Machine Learning to Detect Cyberbullying”. In: *2011 10th International Conference on Machine Learning and Applications and Workshops*. Vol. 2. Honolulu, Hawaii, USA, pp. 241–244. DOI: 10.1109/ICMLA.2011.152.
- Rezvan, Mohammadreza et al. (2018). “A Quality Type-aware Annotated Corpus and Lexicon for Harassment Research”. In: *Proceedings of the 10th ACM Conference on Web Science*. WebSci '18. New York, NY, USA: ACM, pp. 33–36. ISBN: 978-1-

- 4503-5563-6. DOI: 10.1145/3201064.3201103. URL: <http://doi.acm.org/10.1145/3201064.3201103> (visited on 08/08/2019).
- Rosa, H. et al. (Apr. 2019). "Automatic cyberbullying detection: A systematic review". In: *Comput. Hum. Behav.* 93.C, pp. 333–345. ISSN: 0747-5632. DOI: 10.1016/j.chb.2018.12.021. URL: <https://doi.org/10.1016/j.chb.2018.12.021> (visited on 12/05/2024).
- Rudkowsky, Elena et al. (Apr. 2018). "More than Bags of Words: Sentiment Analysis with Word Embeddings". In: *Communication Methods and Measures* 12.2-3, pp. 140–157. ISSN: 1931-2458. DOI: 10.1080/19312458.2018.1455817. URL: <https://doi.org/10.1080/19312458.2018.1455817> (visited on 05/31/2021).
- Rui, Weikang et al. (Mar. 2016). "Unsupervised feature selection for text classification via word embedding". In: *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pp. 1–5. DOI: 10.1109/ICBDA.2016.7509787.
- S. More, Anjali et al. (2018). *Random Forest Classifier Approach for Imbalanced Big Data Classification for Smart City Application Domains*. en. SSRN Scholarly Paper ID 3354727. Rochester, NY: Social Science Research Network. URL: <https://papers.ssrn.com/abstract=3354727> (visited on 11/16/2020).
- Saigal, Pooja et al. (Feb. 2020). "Multi-category news classification using Support Vector Machine based classifiers". en. In: *SN Applied Sciences* 2.3, p. 458. ISSN: 2523-3971. DOI: 10.1007/s42452-020-2266-6. URL: <https://doi.org/10.1007/s42452-020-2266-6> (visited on 07/14/2024).
- Sakaki, Takeshi et al. (2010). "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors". en. In: *WWW2010*. Raleigh, North Carolina, USA, p. 10.
- Salawu, S. et al. (2018). "Approaches to Automated Detection of Cyberbullying: A Survey". In: *IEEE Transactions on Affective Computing*, pp. 1–1. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2017.2761757.
- Santoso, B. et al. (Mar. 2017). "Synthetic Over Sampling Methods for Handling Class Imbalanced Problems : A Review". en. In: *IOP Conference Series: Earth and Envi-*

- ronmental Science* 58, p. 012031. ISSN: 1755-1315. DOI: 10 . 1088 / 1755 - 1315 / 58 / 1 / 012031. URL: <https://doi.org/10.1088%2F1755-1315%2F58%2F1%2F012031> (visited on 10/15/2020).
- Schwartz, H Andrew et al. (2013). "Toward Personality Insights from Language Exploration in Social Media". en. In: SS-13-01, pp. 72–79.
- Seiffert, Chris et al. (Jan. 2010). "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40.1. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, pp. 185–197. ISSN: 1558-2426. DOI: 10.1109/TSMCA.2009.2029559.
- Seliya, N. et al. (Nov. 2009). "A Study on the Relationships of Classifier Performance Metrics". In: *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pp. 59–66. DOI: 10.1109/ICTAI.2009.25.
- Shao, Y. et al. (Dec. 2018). "Clinical Text Classification with Word Embedding Features vs. Bag-of-Words Features". In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2874–2878. DOI: 10.1109/BigData.2018.8622345.
- Sharma, Neha et al. (Dec. 2016). "Modifying Naive Bayes classifier for multinomial text classification". In: *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–7. DOI: 10.1109/ICRAIE.2016.7939519.
- Sharma, Palash et al. (2017). "Challenges and Techniques in Preprocessing for Twitter Data". en. In: *International Journal of Engineering Science and Computing* 7.4, pp. 6611–6613.
- Shu, Jun et al. (Sept. 2019). *Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting*. arXiv:1902.07379 [cs, stat]. DOI: 10.48550/arXiv.1902.07379. URL: <http://arxiv.org/abs/1902.07379> (visited on 07/17/2024).
- Siino, Marco et al. (Mar. 2024). "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers". In: *Information Systems* 121, p. 102342. ISSN: 0306-4379.

- DOI: 10.1016/j.is.2023.102342. URL: <https://www.sciencedirect.com/science/article/pii/S0306437923001783> (visited on 12/03/2024).
- Singh, Gunjan et al. (Jan. 2023). "Text Classification using Improved IWO-HAN". In: *Procedia Computer Science*. International Conference on Machine Learning and Data Engineering 218, pp. 1184–1195. ISSN: 1877-0509. DOI: 10.1016/j.procs.2023.01.097. URL: <https://www.sciencedirect.com/science/article/pii/S1877050923000972> (visited on 12/24/2024).
- Sinoara, Roberta A. et al. (Jan. 2019). "Knowledge-enhanced document embeddings for text classification". en. In: *Knowledge-Based Systems* 163, pp. 955–971. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.10.026. URL: <https://www.sciencedirect.com/science/article/pii/S0950705118305124> (visited on 04/20/2021).
- Slonje, Robert et al. (Apr. 2008). "Cyberbullying: another main type of bullying?" eng. In: *Scandinavian Journal of Psychology* 49.2, pp. 147–154. ISSN: 0036-5564. DOI: 10.1111/j.1467-9450.2007.00611.x.
- Song, Fengxi et al. (Sept. 2005). "A comparative study on text representation schemes in text categorization". en. In: *Pattern Analysis and Applications* 8.1, pp. 199–209. ISSN: 1433-755X. DOI: 10.1007/s10044-005-0256-3. URL: <https://doi.org/10.1007/s10044-005-0256-3> (visited on 05/22/2021).
- Sourander, Andre et al. (July 2010). "Psychosocial Risk Factors Associated With Cyberbullying Among Adolescents: A Population-Based Study". en. In: *Archives of General Psychiatry* 67.7, pp. 720–728. ISSN: 0003-990X. DOI: 10.1001/archgenpsychiatry.2010.79. URL: <https://jamanetwork.com/journals/jamapsychiatry/fullarticle/210833> (visited on 05/12/2020).
- Srinivasan, R. et al. (Mar. 2021). "Sentimental analysis from imbalanced code-mixed data using machine learning approaches". In: *Distributed and Parallel Databases*, pp. 1–16. ISSN: 0926-8782. DOI: 10.1007/s10619-021-07331-4. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7980744/> (visited on 06/05/2021).
- Srividhya, V et al. (2010). "Evaluating Preprocessing Techniques in Text Categorization". In: *International journal of computer science and application* 47.11, pp. 49–51.

- Suhaeni, Cici et al. (Jan. 2023). "Mitigating Class Imbalance in Sentiment Analysis through GPT-3-Generated Synthetic Sentences". en. In: *Applied Sciences* 13.17. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute, p. 9766. ISSN: 2076-3417. DOI: 10.3390/app13179766. URL: <https://www.mdpi.com/2076-3417/13/17/9766> (visited on 09/22/2023).
- Sumner, Chris et al. (Dec. 2012). "Predicting Dark Triad Personality Traits from Twitter Usage and a Linguistic Analysis of Tweets". In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. Boca Raton, Florida, USA, pp. 386–393. DOI: 10.1109/ICMLA.2012.218.
- Sun, Aixin et al. (2009). "On strategies for imbalanced text classification using SVM: A comparative study. Decision Support Systems". In: *Decision Support Systems* 40.1, pp. 191–201.
- Sun, Fei et al. (July 2019). "Efficiency of Extreme Gradient Boosting for Imbalanced Land Cover Classification Using an Extended Margin and Disagreement Performance". en. In: *ISPRS International Journal of Geo-Information* 8.7, p. 315. DOI: 10.3390/ijgi8070315. URL: <https://www.mdpi.com/2220-9964/8/7/315> (visited on 11/17/2020).
- Sun, Y. et al. (Dec. 2006). "Boosting for Learning Multiple Classes with Imbalanced Class Distribution". In: *Sixth International Conference on Data Mining (ICDM'06)*, pp. 592–602. DOI: 10.1109/ICDM.2006.29.
- Sun, Yanmin et al. (June 2009). "Classification of imbalanced data: A review." en. In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.04, pp. 687–719. ISSN: 0218-0014, 1793-6381. DOI: 10.1142/S0218001409007326. URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218001409007326> (visited on 05/11/2020).
- Tahir, Mirza Amaad Ul Haq et al. (2019). "A Classification Model For Class Imbalance Dataset Using Genetic Programming". In: *IEEE Access* 7, pp. 71013–71037. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2915611.

- Tahir, Muhammad Atif et al. (2009). "A Multiple Expert Approach to the Class Imbalance Problem Using Inverse Random under Sampling". en. In: *Multiple Classifier Systems*. Ed. by Jón Atli Benediktsson et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 82–91. ISBN: 978-3-642-02326-2.
- Talpur, Bandeh Ali et al. (Oct. 2020). "Cyberbullying severity detection: A machine learning approach". en. In: *PLOS ONE* 15.10, e0240924. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0240924. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0240924> (visited on 10/30/2020).
- Tan, S et al. (May 2008). "An empirical study of sentiment analysis for chinese documents". en. In: *Expert Systems with Applications* 34.4, pp. 2622–2629. ISSN: 09574174. DOI: 10.1016/j.eswa.2007.05.028. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0957417407001534> (visited on 10/02/2023).
- Tang, Mingzhu et al. (July 2014). "Cost-Sensitive Support Vector Machine Using Randomized Dual Coordinate Descent Method for Big Class-Imbalanced Data Classification". en. In: *Abstract and Applied Analysis* 2014, e416591. ISSN: 1085-3375. DOI: 10.1155/2014/416591. URL: <https://www.hindawi.com/journals/aaa/2014/416591/> (visited on 03/24/2021).
- Tanha, Jafar et al. (Sept. 2020). "Boosting methods for multi-class imbalanced data classification: an experimental review". en. In: *Journal of Big Data* 7.1, p. 70. ISSN: 2196-1115. DOI: 10.1186/s40537-020-00349-y. URL: <https://doi.org/10.1186/s40537-020-00349-y> (visited on 11/17/2020).
- Teh, Kevin et al. (Dec. 2020). "Imbalanced learning: Improving classification of diabetic neuropathy from magnetic resonance imaging". en. In: *PLOS ONE* 15.12, e0243907. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0243907. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0243907> (visited on 04/01/2021).
- Toman, Michal et al. (2006). "Influence of Word Normalization on Text Classification". en. In: *Proceedings of InSciT*, pp. 354–358.

- Toolan, Fergus et al. (2010). "Feature selection for Spam and Phishing detection". In: *2010 eCrime Researchers Summit*. Dallas, USA: IEEE, pp. 1–12. ISBN: 978-1-4244-7760-9 978-1-4244-7761-6. DOI: 10.1109/ecrime.2010.5706696. URL: <http://ieeexplore.ieee.org/document/5706696/> (visited on 10/01/2023).
- Turney, P. D. (Feb. 1995). "Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm". In: *arXiv:cs/9503102* 2, pp. 369–409. URL: <http://arxiv.org/abs/cs/9503102> (visited on 03/27/2021).
- Turney, Peter (July 2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 417–424. DOI: 10.3115/1073083.1073153. URL: <https://www.aclweb.org/anthology/P02-1053> (visited on 09/08/2019).
- Turney, Peter D. et al. (Oct. 2003). "Measuring praise and criticism: Inference of semantic orientation from association". en. In: *ACM Transactions on Information Systems* 21.4, pp. 315–346. ISSN: 1046-8188, 1558-2868. DOI: 10.1145/944012.944013. URL: <https://dl.acm.org/doi/10.1145/944012.944013> (visited on 08/08/2023).
- Uysal, Alper Kursat et al. (Jan. 2014). "The impact of preprocessing on text classification". en. In: *Information Processing & Management* 50.1, pp. 104–112. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2013.08.006. URL: <https://www.sciencedirect.com/science/article/pii/S0306457313000964> (visited on 04/28/2021).
- Van Hulse, Jason et al. (June 2007). "Experimental perspectives on learning from imbalanced data". en. In: *Proceedings of the 24th international conference on Machine learning*. Corvalis Oregon USA: ACM, pp. 935–942. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273614. URL: <https://dl.acm.org/doi/10.1145/1273496.1273614> (visited on 08/24/2023).
- Van Royen, Kathleen et al. (Feb. 2015). "Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability". en. In:

- Telematics and Informatics* 32.1, pp. 89–97. ISSN: 0736-5853. DOI: 10.1016/j.tele.2014.04.002. URL: <http://www.sciencedirect.com/science/article/pii/S073658531400032X> (visited on 05/12/2020).
- Vieira, Susana M. et al. (July 2010). “Cohen’s kappa coefficient as a performance measure for feature selection”. In: *International Conference on Fuzzy Systems*. ISSN: 1098-7584, pp. 1–8. DOI: 10.1109/FUZZY.2010.5584447.
- Vijayakumar, Ashwin et al. (Sept. 2017). “Sound-Word2Vec: Learning Word Representations Grounded in Sounds”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 920–925. DOI: 10.18653/v1/D17-1096. URL: <https://www.aclweb.org/anthology/D17-1096> (visited on 04/22/2021).
- Wah, Yap Bee et al. (2016). “Handling imbalanced dataset using SVM and k-NN approach”. en. In: Johor Bahru, Malaysia, p. 020023. DOI: 10.1063/1.4954536. URL: <http://aip.scitation.org/doi/abs/10.1063/1.4954536> (visited on 11/16/2020).
- Wang, Huanjing et al. (Feb. 2015). “An Empirical Investigation on Wrapper-Based Feature Selection for Predicting Software Quality”. In: *International Journal of Software Engineering and Knowledge Engineering* 25.01. Publisher: World Scientific Publishing Co., pp. 93–114. ISSN: 0218-1940. DOI: 10.1142/S0218194015400057. URL: <https://www.worldscientific.com/doi/10.1142/S0218194015400057> (visited on 08/24/2023).
- Wang, Shuo (2011). “Ensemble diversity for class imbalance learning”. English. d\_ph. University of Birmingham. URL: <https://etheses.bham.ac.uk/id/eprint/1793/> (visited on 04/14/2021).
- Wang, Shuo and Xin Yao (Aug. 2012). “Multiclass Imbalance Problems: Analysis and Potential Solutions”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4, pp. 1119–1130. ISSN: 1941-0492. DOI: 10.1109/TSMCB.2012.2187280.

- Wasikowski, Mike et al. (Oct. 2010). “Combating the Small Sample Class Imbalance Problem Using Feature Selection”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10. Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 1388–1400. ISSN: 1558-2191. DOI: 10.1109/TKDE.2009.187.
- Wei, Jason et al. (Nov. 2019). “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 6382–6388. DOI: 10.18653/v1/D19-1670. URL: <https://aclanthology.org/D19-1670> (visited on 07/16/2024).
- Weiss, Gary M. et al. (Oct. 2003). “Learning when training data are costly: the effect of class distribution on tree induction: Journal of Artificial Intelligence Research: Vol 19, No 1”. In: 19.1, pp. 315–354. URL: <https://dl.acm.org/doi/10.5555/1622434.1622445> (visited on 08/18/2023).
- Whissell, Cynthia (Oct. 2009). “Using the Revised Dictionary of Affect in Language to quantify the emotional undertones of samples of natural language”. eng. In: *Psychological Reports* 105.2, pp. 509–521. ISSN: 0033-2941. DOI: 10.2466/PRO.105.2.509-521.
- Wu, Shun (Aug. 2018). “Short Text Mining for Fault Diagnosis of Railway System Based on Multi-Granularity Topic Model”. In: *2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS)*, pp. 1–6. DOI: 10.1109/LISS.2018.8593228.
- Wu, Xing et al. (2019). “Conditional BERT Contextual Augmentation”. en. In: *Computational Science – ICCS 2019*. Ed. by João M. F. Rodrigues et al. Cham: Springer International Publishing, pp. 84–95. ISBN: 978-3-030-22747-0. DOI: 10.1007/978-3-030-22747-0\_7.
- Xing, Eric P. et al. (Jan. 2002). “Distance metric learning, with application to clustering with side-information”. In: *Proceedings of the 15th International Conference on*

- Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, pp. 521–528. (Visited on 08/24/2023).
- Yan, Dongyang et al. (2020). “Network-Based Bag-of-Words Model for Text Classification”. In: *IEEE Access* 8, pp. 82641–82652. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2991074.
- Yang, Jieming et al. (2014). “Improved Feature-Selection Method Considering the Imbalance Problem in Text Categorization”. In: *The Scientific World Journal* 2014. ISSN: 2356-6140. DOI: 10.1155/2014/625342. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4058251/> (visited on 08/17/2020).
- Yang, Pengyi et al. (2013). “Ensemble-Based Wrapper Methods for Feature Selection and Class Imbalance Learning”. en. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 544–555. ISBN: 978-3-642-37453-1. DOI: 10.1007/978-3-642-37453-1\_45.
- Yang, Zhen et al. (May 2022). “A systematic literature review of methods and datasets for anomaly-based network intrusion detection”. In: *Computers & Security* 116, p. 102675. ISSN: 0167-4048. DOI: 10.1016/j.cose.2022.102675. URL: <https://www.sciencedirect.com/science/article/pii/S0167404822000736> (visited on 12/09/2024).
- Yeh, Jen-Yuan et al. (Oct. 2021). “FEATURE SELECTION AND CLASSIFICATION INTEGRATED METHOD FOR IDENTIFYING CITED TEXT SPANS FOR CITANCES ON IMBALANCED DATA”. en. In: *Malaysian Journal of Computer Science* 34.4. Number: 4, pp. 355–373. ISSN: 0127-9084. DOI: 10.22452/mjcs.vol34no4.3. URL: <https://ejournal.um.edu.my/index.php/MJCS/article/view/19799> (visited on 10/02/2023).
- Yi Liu et al. (2005). “One-against-all multi-class SVM classification using reliability measures”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*. Vol. 2. Montreal, Que., Canada: IEEE, pp. 849–854. ISBN: 978-0-

- 7803-9048-5. DOI: 10.1109/IJCNN.2005.1555963. URL: <http://ieeexplore.ieee.org/document/1555963/> (visited on 08/29/2019).
- Yin, Liuzhi et al. (Apr. 2013). "Feature selection for high-dimensional imbalanced data". en. In: *Neurocomputing*. Learning for Scalable Multimedia Representation 105, pp. 3–11. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2012.04.039. URL: <https://www.sciencedirect.com/science/article/pii/S0925231212007126> (visited on 03/24/2021).
- Yu, S. et al. (2019). "The OCS-SVM: An Objective-Cost-Sensitive SVM With Sample-Based Misclassification Cost Invariance". In: *IEEE Access* 7, pp. 118931–118942. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2933437.
- Zeng, Tao et al. (May 2015). "Deep convolutional neural networks for annotating gene expression patterns in the mouse brain". en. In: *BMC Bioinformatics* 16.1, p. 147. ISSN: 1471-2105. DOI: 10.1186/s12859-015-0553-9. URL: <https://doi.org/10.1186/s12859-015-0553-9> (visited on 04/23/2021).
- Zhang, C. et al. (Dec. 2017). "Feature selection for high dimensional imbalanced class data based on F-measure optimization". In: *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 278–283. DOI: 10.1109/SPAC.2017.8304290.
- Zhang, Xingfu et al. (May 2021). "Threshold Moving Approaches for Addressing the Class Imbalance Problem and their Application to Multi-label Classification". In: *Proceedings of the 4th International Conference on Advances in Image Processing*. ICAIP '20. New York, NY, USA: Association for Computing Machinery, pp. 72–77. ISBN: 978-1-4503-8836-8. DOI: 10.1145/3441250.3441274. URL: <https://doi.org/10.1145/3441250.3441274> (visited on 07/17/2024).
- Zhao, R. et al. (July 2017). "Cyberbullying Detection Based on Semantic-Enhanced Marginalized Denoising Auto-Encoder". In: *IEEE Transactions on Affective Computing* 8.3, pp. 328–339. DOI: 10.1109/TAFFC.2016.2531682.
- Zhao, Rui et al. (2016). "Automatic Detection of Cyberbullying on Social Networks Based on Bullying Features". In: *Proceedings of the 17th International Conference on*

- Distributed Computing and Networking*. ICDCN '16. New York, NY, USA: ACM, 43:1–43:6. ISBN: 978-1-4503-4032-8. DOI: 10.1145/2833312.2849567. URL: <http://doi.acm.org/10.1145/2833312.2849567> (visited on 08/01/2019).
- Zhao, Yang et al. (May 2018). *A Framework of Rebalancing Imbalanced Healthcare Data for Rare Events' Classification: A Case of Look-Alike Sound-Alike Mix-Up Incident Detection*. en. Research Article. URL: <https://www.hindawi.com/journals/jhe/2018/6275435/> (visited on 08/17/2020).
- Zhen, Liu et al. (Jan. 2012). "A New Feature Selection Method for Internet Traffic Classification Using ML". en. In: *Physics Procedia*. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012) 33, pp. 1338–1345. ISSN: 1875-3892. DOI: 10.1016/j.phpro.2012.05.220. URL: <https://www.sciencedirect.com/science/article/pii/S1875389212015337> (visited on 03/24/2021).
- Zheng, Zhaohui et al. (June 2004). "Feature selection for text categorization on imbalanced data". In: *ACM SIGKDD Explorations Newsletter* 6.1, pp. 80–89. ISSN: 1931-0145. DOI: 10.1145/1007730.1007741. URL: <https://doi.org/10.1145/1007730.1007741> (visited on 10/06/2023).
- Zhou, Yanling et al. (2020). "Deep Learning Based Fusion Approach for Hate Speech Detection". In: *IEEE Access* 8. Conference Name: IEEE Access, pp. 128923–128929. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3009244.
- Zsila, Ágnes et al. (Oct. 2019). "Gender Differences in the Association Between Cyberbullying Victimization and Perpetration: The Role of Anger Rumination and Traditional Bullying Experiences". en. In: *International Journal of Mental Health and Addiction* 17.5, pp. 1252–1267. ISSN: 1557-1874, 1557-1882. DOI: 10.1007/s11469-018-9893-9. URL: <http://link.springer.com/10.1007/s11469-018-9893-9> (visited on 05/09/2020).